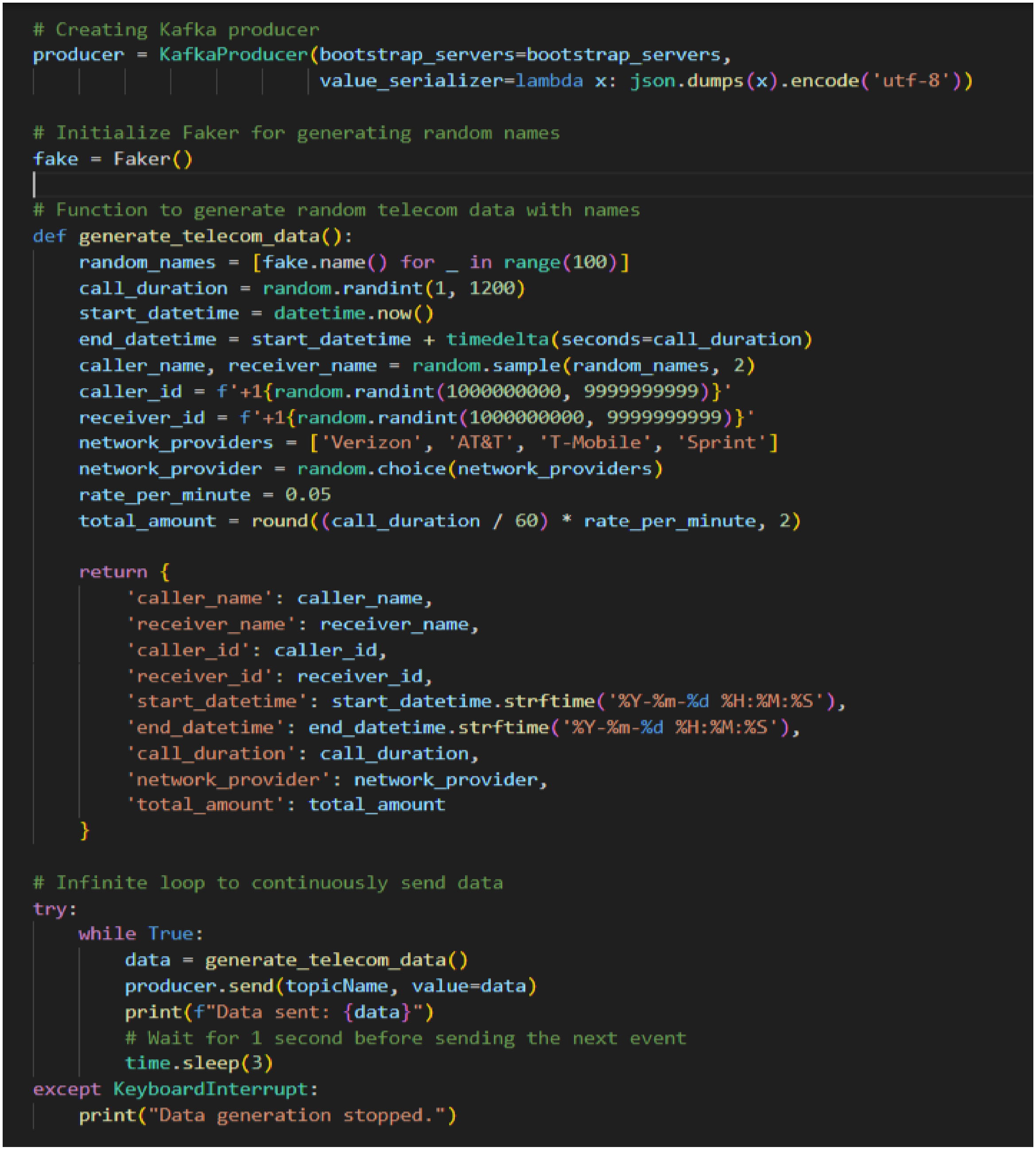
Real-Time Telecommunication Data Processing (Kafka-Redshift)

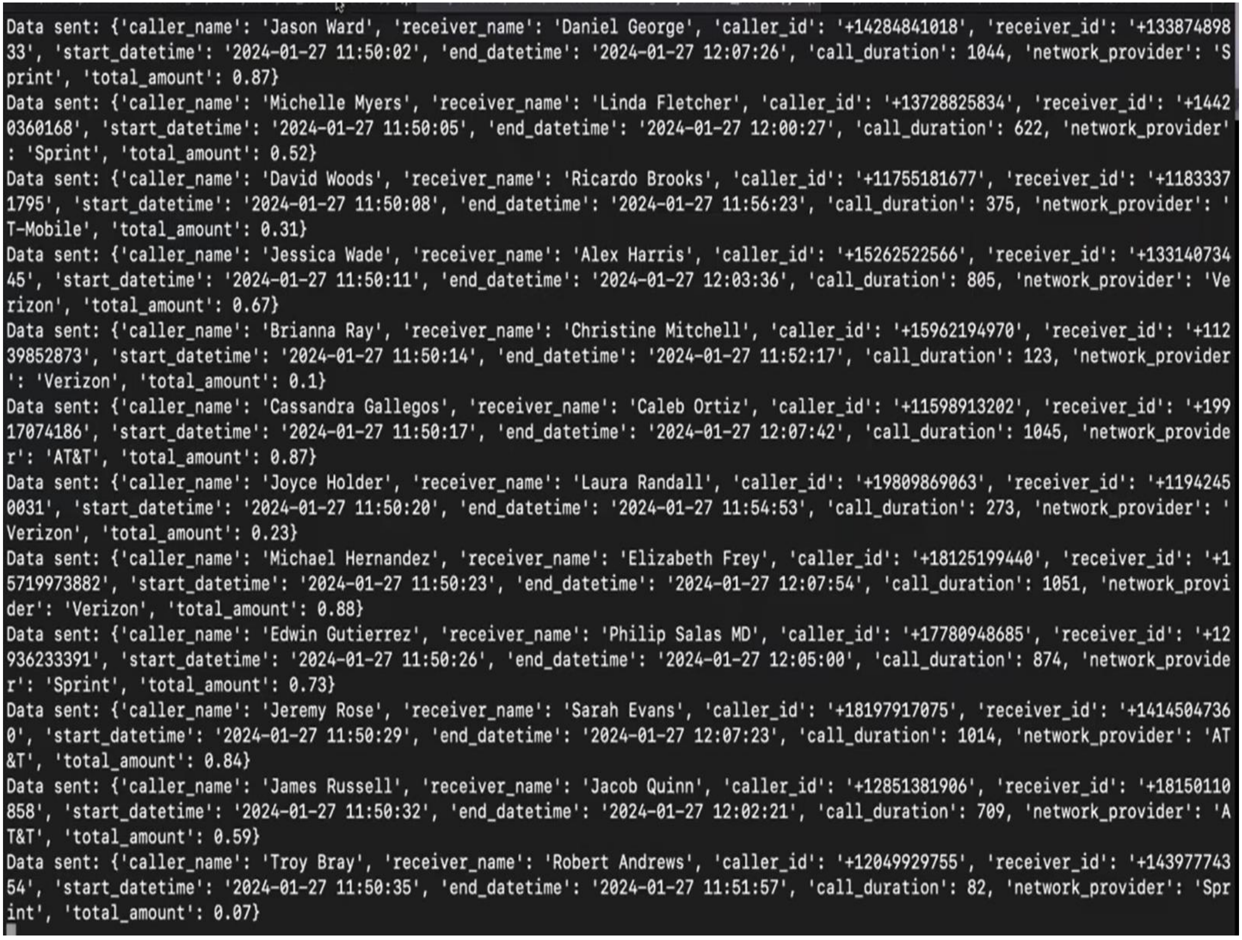
1. We had a producer script which was dumping telecommunication data on kafka topic. Then from there we had a streaming application which doing some filtrations and ingesting data in real time at redshift warehouse. We dumped data into warehouse & we had a quick sight dashboard on the top of it.

producer script:



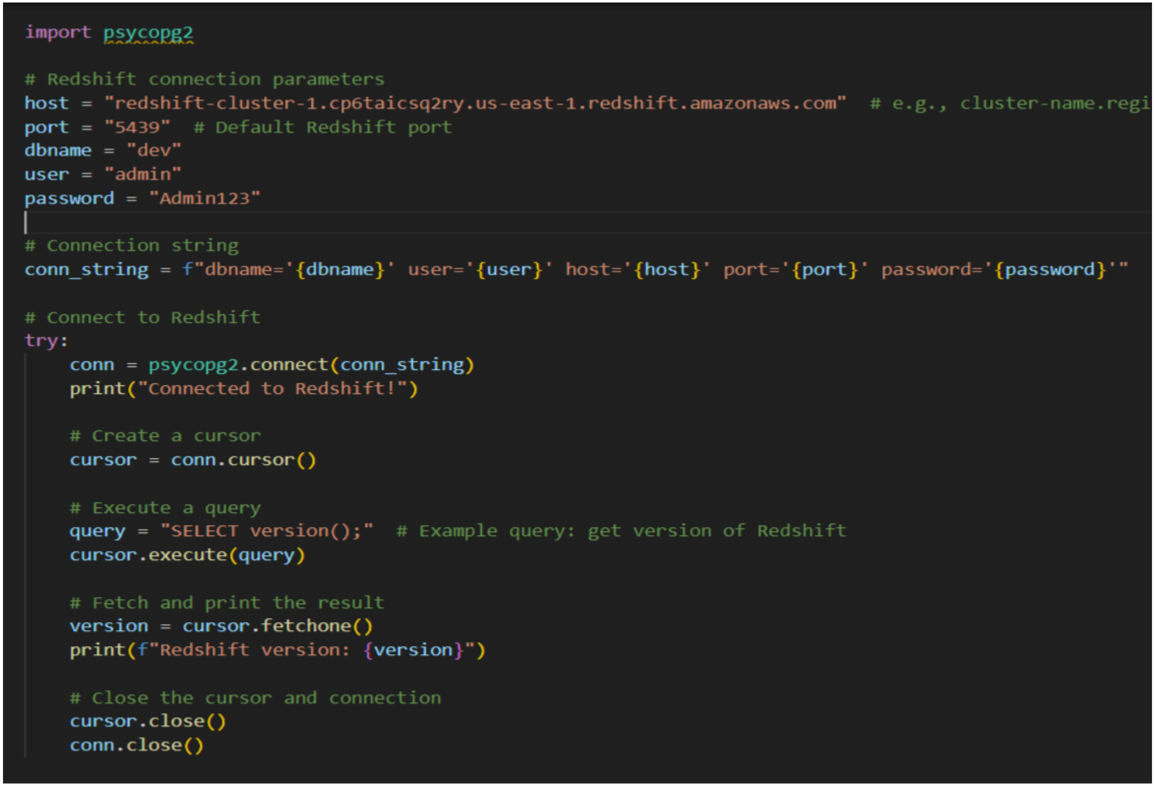
```python
# Creating Kafka producer
producer = KafkaProducer(bootstrap_servers=bootstrap_servers,
                         value_serializer=lambda x: json.dumps(x).encode('utf-8'))

# Initialize Faker for generating random names
fake = Faker()

# Function to generate random telecom data with names
def generate_telecom_data():
    random_names = [fake.name() for _ in range(100)]
    call_duration = random.randint(1, 1200)
    start_datetime = datetime.now()
    end_datetime = start_datetime + timedelta(seconds=call_duration)
    caller_name, receiver_name = random.sample(random_names, 2)
    caller_id = f'+1{random.randint(1000000000, 9999999999)}'
    receiver_id = f'+1{random.randint(1000000000, 9999999999)}'
    network_providers = ['Verizon', 'AT&T', 'T-Mobile', 'Sprint']
    network_provider = random.choice(network_providers)
    rate_per_minute = 0.05
    total_amount = round((call_duration / 60) * rate_per_minute, 2)

    return {
        'caller_name': caller_name,
        'receiver_name': receiver_name,
        'caller_id': caller_id,
        'receiver_id': receiver_id,
        'start_datetime': start_datetime.strftime('%Y-%m-%d %H:%M:%S'),
        'end_datetime': end_datetime.strftime('%Y-%m-%d %H:%M:%S'),
        'call_duration': call_duration,
        'network_provider': network_provider,
        'total_amount': total_amount
    }

# Infinite loop to continuously send data
try:
    while True:
        data = generate_telecom_data()
        producer.send(topicName, value=data)
        print(f"Data sent: {data}")
        # Wait for 1 second before sending the next event
        time.sleep(3)
except KeyboardInterrupt:
    print("Data generation stopped.")
```

data getting produced to kafka topic:



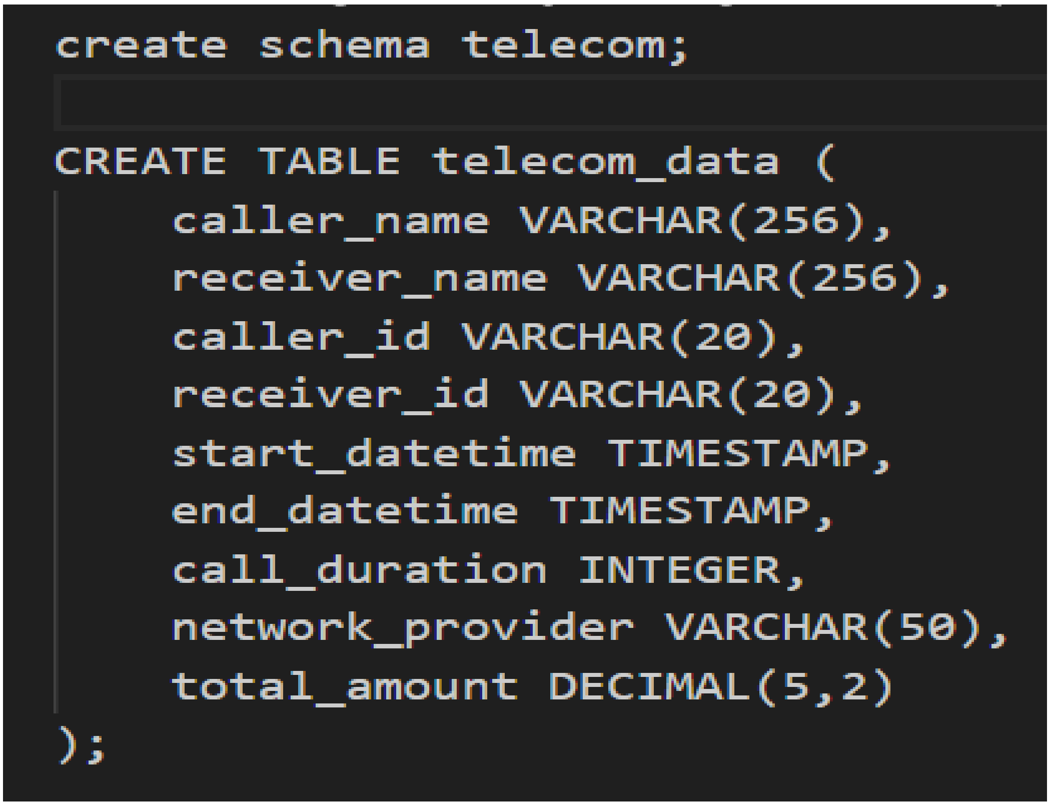
```
Data sent: {'caller_name': 'Jason Ward', 'receiver_name': 'Daniel George', 'caller_id': '+14284841018', 'receiver_id': '+133874898
33', 'start_datetime': '2024-01-27 11:50:02', 'end_datetime': '2024-01-27 12:07:26', 'call_duration': 1044, 'network_provider': 'S
print', 'total_amount': 0.87}
Data sent: {'caller_name': 'Michelle Myers', 'receiver_name': 'Linda Fletcher', 'caller_id': '+13728825834', 'receiver_id': '+1442
0360168', 'start_datetime': '2024-01-27 11:50:05', 'end_datetime': '2024-01-27 12:00:27', 'call_duration': 622, 'network_provider'
: 'Sprint', 'total_amount': 0.52}
Data sent: {'caller_name': 'David Woods', 'receiver_name': 'Ricardo Brooks', 'caller_id': '+11755181677', 'receiver_id': '+1183337
1795', 'start_datetime': '2024-01-27 11:50:08', 'end_datetime': '2024-01-27 11:56:23', 'call_duration': 375, 'network_provider': '
T-Mobile', 'total_amount': 0.31}
Data sent: {'caller_name': 'Jessica Wade', 'receiver_name': 'Alex Harris', 'caller_id': '+15262522566', 'receiver_id': '+133140734
45', 'start_datetime': '2024-01-27 11:50:11', 'end_datetime': '2024-01-27 12:03:36', 'call_duration': 805, 'network_provider': 'Ve
rizon', 'total_amount': 0.67}
Data sent: {'caller_name': 'Brianna Ray', 'receiver_name': 'Christine Mitchell', 'caller_id': '+15962194970', 'receiver_id': '+112
39852873', 'start_datetime': '2024-01-27 11:50:14', 'end_datetime': '2024-01-27 11:52:17', 'call_duration': 123, 'network_provider
': 'Verizon', 'total_amount': 0.1}
Data sent: {'caller_name': 'Cassandra Gallegos', 'receiver_name': 'Caleb Ortiz', 'caller_id': '+11598913202', 'receiver_id': '+199
17074186', 'start_datetime': '2024-01-27 11:50:17', 'end_datetime': '2024-01-27 12:07:42', 'call_duration': 1045, 'network_provide
r': 'AT&T', 'total_amount': 0.87}
Data sent: {'caller_name': 'Joyce Holder', 'receiver_name': 'Laura Randall', 'caller_id': '+19809869063', 'receiver_id': '+1194245
0031', 'start_datetime': '2024-01-27 11:50:20', 'end_datetime': '2024-01-27 11:54:53', 'call_duration': 273, 'network_provider': '
Verizon', 'total_amount': 0.23}
Data sent: {'caller_name': 'Michael Hernandez', 'receiver_name': 'Elizabeth Frey', 'caller_id': '+18125199440', 'receiver_id': '+1
5719973882', 'start_datetime': '2024-01-27 11:50:23', 'end_datetime': '2024-01-27 12:07:54', 'call_duration': 1051, 'network_provi
der': 'Verizon', 'total_amount': 0.88}
Data sent: {'caller_name': 'Edwin Gutierrez', 'receiver_name': 'Philip Salas MD', 'caller_id': '+17780948685', 'receiver_id': '+12
936233391', 'start_datetime': '2024-01-27 11:50:26', 'end_datetime': '2024-01-27 12:05:00', 'call_duration': 874, 'network_provide
r': 'Sprint', 'total_amount': 0.73}
Data sent: {'caller_name': 'Jeremy Rose', 'receiver_name': 'Sarah Evans', 'caller_id': '+18197917075', 'receiver_id': '+1414504736
0', 'start_datetime': '2024-01-27 11:50:29', 'end_datetime': '2024-01-27 12:07:23', 'call_duration': 1014, 'network_provider': 'AT
&T', 'total_amount': 0.84}
Data sent: {'caller_name': 'James Russell', 'receiver_name': 'Jacob Quinn', 'caller_id': '+12851381906', 'receiver_id': '+18150110
858', 'start_datetime': '2024-01-27 11:50:32', 'end_datetime': '2024-01-27 12:02:21', 'call_duration': 709, 'network_provider': 'A
T&T', 'total_amount': 0.59}
Data sent: {'caller_name': 'Troy Bray', 'receiver_name': 'Robert Andrews', 'caller_id': '+12049929755', 'receiver_id': '+143977743
54', 'start_datetime': '2024-01-27 11:50:35', 'end_datetime': '2024-01-27 11:51:57', 'call_duration': 82, 'network_provider': 'Spr
int', 'total_amount': 0.07}
```

2. we created inbound rule like from where the inbound traffic can come to the port. So actually we connected our local script with the aws redshift.



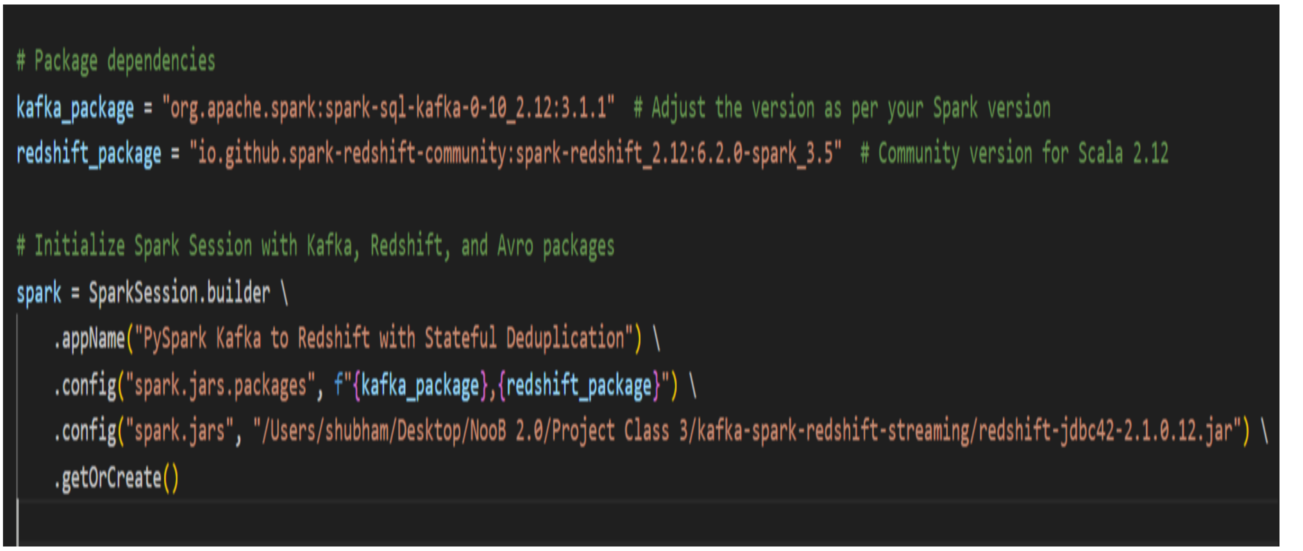
```python
import psycopg2

# Redshift connection parameters
host = "redshift-cluster-1.cp6taicsq2ry.us-east-1.redshift.amazonaws.com"  # e.g., cluster-name.regi
port = "5439"  # Default Redshift port
dbname = "dev"
user = "admin"
password = "Admin123"

# Connection string
conn_string = f"dbname='{dbname}' user='{user}' host='{host}' port='{port}' password='{password}'"

# Connect to Redshift
try:
    conn = psycopg2.connect(conn_string)
    print("Connected to Redshift!")

    # Create a cursor
    cursor = conn.cursor()

    # Execute a query
    query = "SELECT version();"  # Example query: get version of Redshift
    cursor.execute(query)

    # Fetch and print the result
    version = cursor.fetchone()
    print(f"Redshift version: {version}")

    # Close the cursor and connection
    cursor.close()
    conn.close()
```

3. On the other hand, we created then redshift schema and table where we will be ingesting the data.



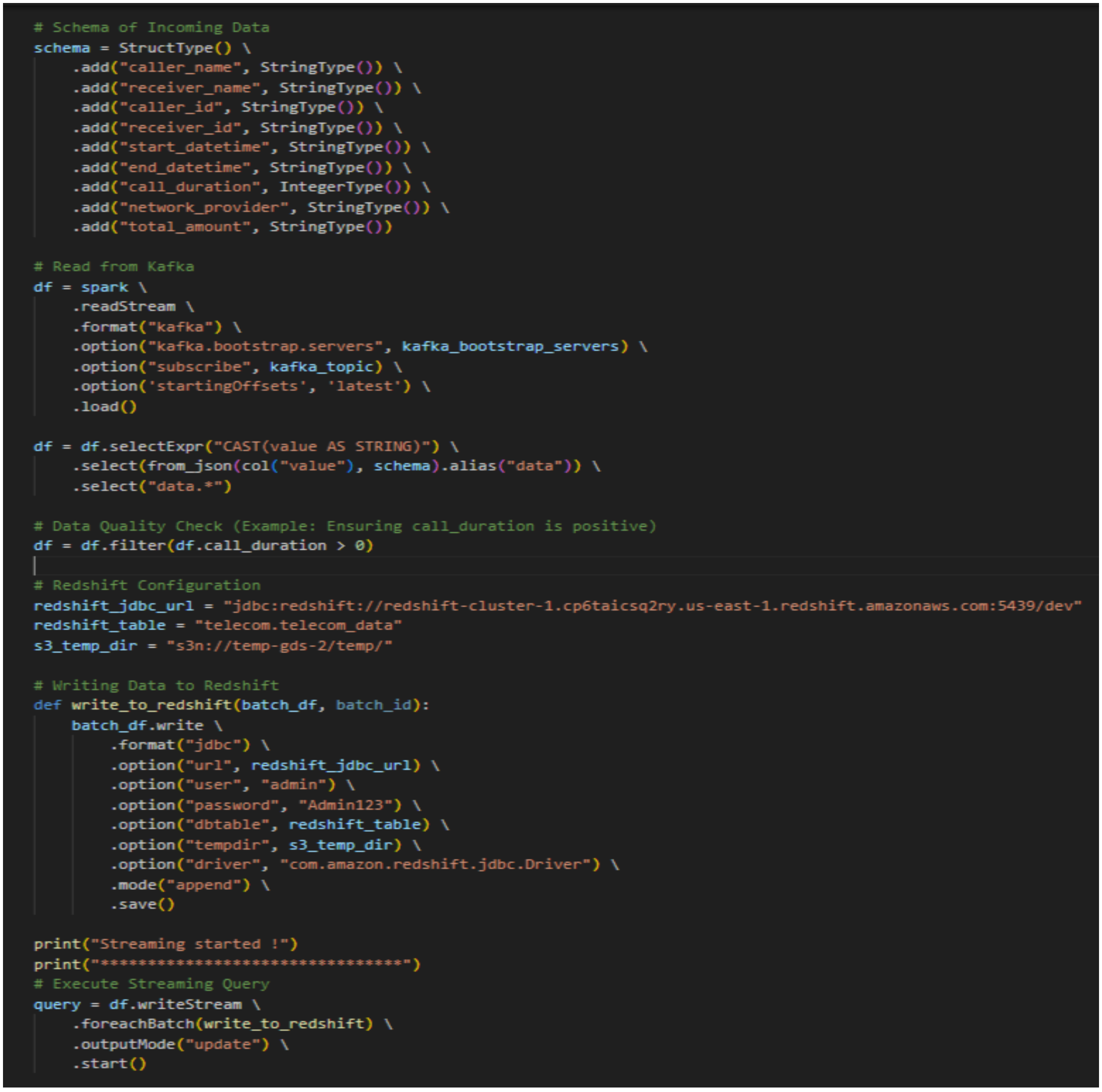
```sql
create schema telecom;

CREATE TABLE telecom_data (
    caller_name VARCHAR(256),
    receiver_name VARCHAR(256),
    caller_id VARCHAR(20),
    receiver_id VARCHAR(20),
    start_datetime TIMESTAMP,
    end_datetime TIMESTAMP,
    call_duration INTEGER,
    network_provider VARCHAR(50),
    total_amount DECIMAL(5,2)
);
```

4. Now we had streaming app which will be connecting to kafka and redshift so we included kafka-spark & redshift-spark connector jar file respectively. We also needed redshift jdbs connector because we were writing data in jdbc format as per our spark version



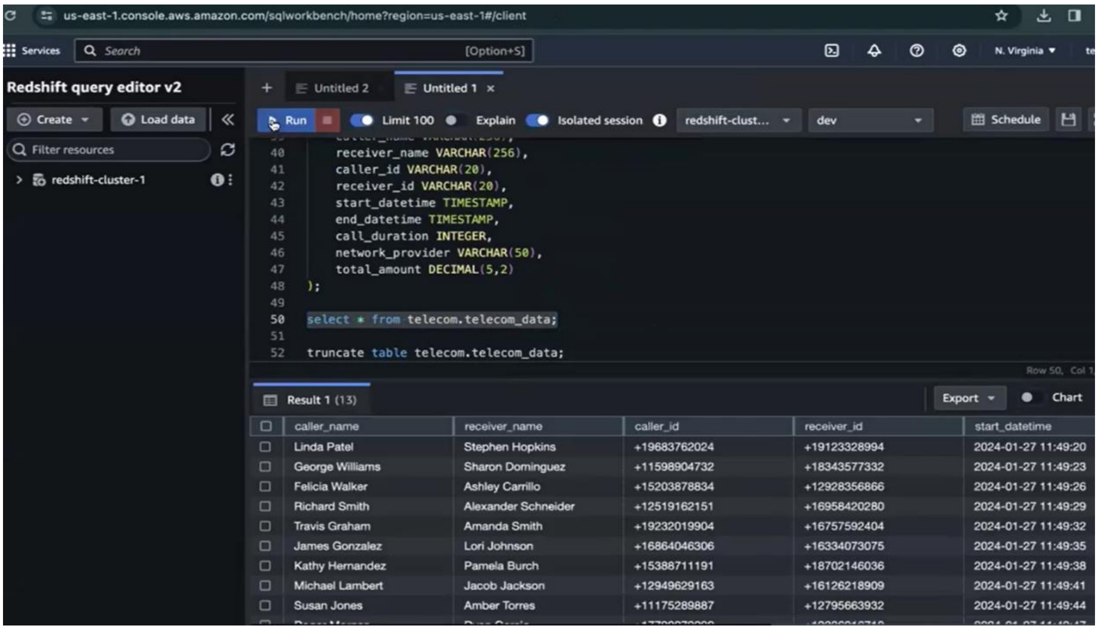
```python
# Package dependencies
kafka_package = "org.apache.spark:spark-sql-kafka-0-10_2.12:3.1.1"  # Adjust the version as per your Spark version
redshift_package = "io.github.spark-redshift-community:spark-redshift_2.12:6.2.0-spark_3.5"  # Community version for Scala 2.12

# Initialize Spark Session with Kafka, Redshift, and Avro packages
spark = SparkSession.builder \
    .appName("PySpark Kafka to Redshift with Stateful Deduplication") \
    .config("spark.jars.packages", f"{kafka_package},{redshift_package}") \
    .config("spark.jars", "/Users/shubham/Desktop/NooB 2.0/Project Class 3/kafka-spark-redshift-streaming/redshift-jdbc42-2.1.0.12.jar") \
    .getOrCreate()
```

5. We created next schema of incoming data from kafka topic and used readStream to read data from kafka topic. Hence did some transformations and writing data to redshift using writeStream.



```
# Schema of Incoming Data
schema = StructType() \
    .add("caller_name", StringType()) \
    .add("receiver_name", StringType()) \
    .add("caller_id", StringType()) \
    .add("receiver_id", StringType()) \
    .add("start_datetime", StringType()) \
    .add("end_datetime", StringType()) \
    .add("call_duration", IntegerType()) \
    .add("network_provider", StringType()) \
    .add("total_amount", StringType())

# Read from Kafka
df = spark \
    .readStream \
    .format("kafka") \
    .option("kafka.bootstrap.servers", kafka_bootstrap_servers) \
    .option("subscribe", kafka_topic) \
    .option('startingOffsets', 'latest') \
    .load()

df = df.selectExpr("CAST(value AS STRING)") \
    .select(from_json(col("value"), schema).alias("data")) \
    .select("data.*")

# Data Quality Check (Example: Ensuring call_duration is positive)
df = df.filter(df.call_duration > 0)

# Redshift Configuration
redshift_jdbc_url = "jdbc:redshift://redshift-cluster-1.cp6taicsq2ry.us-east-1.redshift.amazonaws.com:5439/dev"
redshift_table = "telecom.telecom_data"
s3_temp_dir = "s3n://temp-gds-2/temp/"

# Writing Data to Redshift
def write_to_redshift(batch_df, batch_id):
    batch_df.write \
        .format("jdbc") \
        .option("url", redshift_jdbc_url) \
        .option("user", "admin") \
        .option("password", "Admin123") \
        .option("dbtable", redshift_table) \
        .option("tempdir", s3_temp_dir) \
        .option("driver", "com.amazon.redshift.jdbc.Driver") \
        .mode("append") \
        .save()

print("Streaming started !")
print("***********************************")
# Execute Streaming Query
query = df.writeStream \
    .foreachBatch(write_to_redshift) \
    .outputMode("update") \
    .start()
```

Redshift table results:

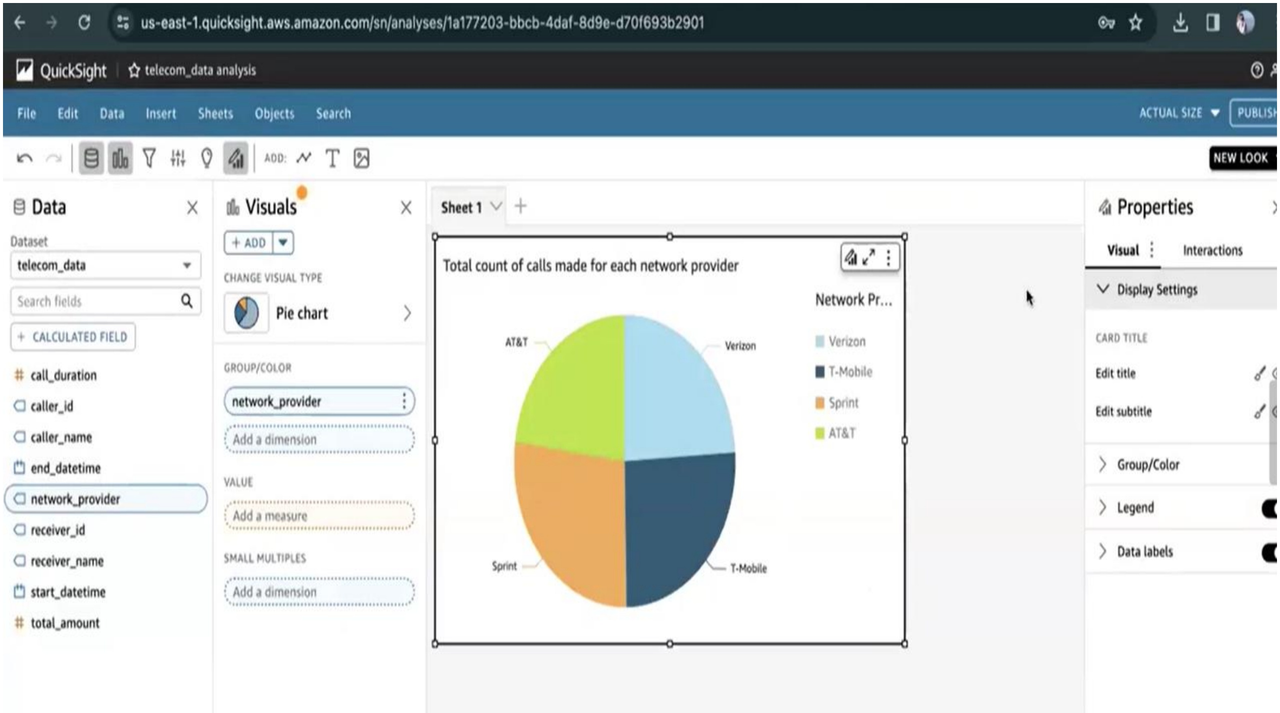


```
    receiver_name VARCHAR(256),
    caller_id VARCHAR(20),
    receiver_id VARCHAR(20),
    start_datetime TIMESTAMP,
    end_datetime TIMESTAMP,
    call_duration INTEGER,
    network_provider VARCHAR(50),
    total_amount DECIMAL(5,2)
);

select * from telecom.telecom_data;

truncate table telecom.telecom_data;
```

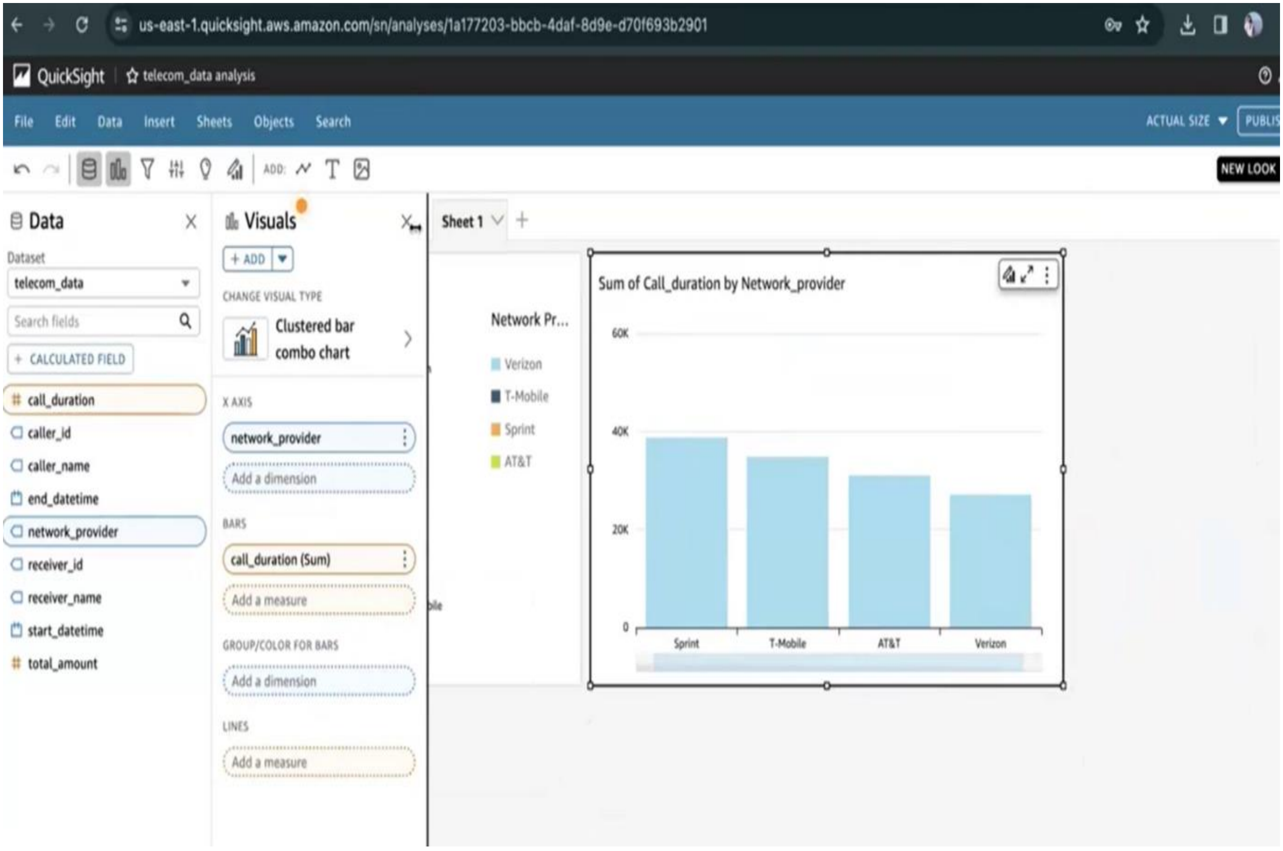
caller_name	receiver_name	caller_id	receiver_id	start_datetime
Linda Patel	Stephen Hopkins	+19683762024	+19123328994	2024-01-27 11:49:20
George Williams	Sharon Dominguez	+11598904732	+18343577332	2024-01-27 11:49:23
Felicia Walker	Ashley Carrillo	+15203878834	+12928356866	2024-01-27 11:49:26
Richard Smith	Alexander Schneider	+12519162151	+16958420280	2024-01-27 11:49:29
Travis Graham	Amanda Smith	+19232019904	+16757592404	2024-01-27 11:49:32
James Gonzalez	Lori Johnson	+16864046306	+16334073075	2024-01-27 11:49:35
Kathy Hernandez	Pamela Burch	+15388711191	+18702146036	2024-01-27 11:49:38
Michael Lambert	Jacob Jackson	+12949629163	+16126218909	2024-01-27 11:49:41
Susan Jones	Amber Torres	+11175289887	+12795663932	2024-01-27 11:49:44

6. next we worked on aws quicksight. Created dataset and made dashboard on the top of it like network provider wise calls, network provider wise total call duration

network provider wise calls:



network provider wise total call duration:


us-east-1.quicksight.aws.amazon.com/sn/analyses/1a177203-bbcb-4daf-8d9e-d70f693b2901
QuickSight
telecom_data analysis
File
Edit
Data
Insert
Sheets
Objects
Search
ACTUAL SIZE
PUBLIS
ADD:
NEW LOOK
Data
Dataset
telecom_data
Search fields
+ CALCULATED FIELD
call_duration
caller_id
caller_name
end_datetime
network_provider
receiver_id
receiver_name
start_datetime
total_amount
Visuals
+ ADD
CHANGE VISUAL TYPE
Clustered bar combo chart
X AXIS
network_provider
Add a dimension
BARS
call_duration (Sum)
Add a measure
GROUP/COLOR FOR BARS
Add a dimension
LINES
Add a measure
Sheet 1
Network Pr...
Verizon
T-Mobile
Sprint
AT&T
Sum of Call_duration by Network_provider
60K
40K
20K
0
Sprint
T-Mobile
AT&T
Verizon