```
# 00_explore_inter_trade_time.ipynb

# Import necessary PySpark functions
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, to_date, lag, datediff, avg, count
from pyspark.sql.window import Window
import os

# --- 0. Mount Google Drive (if using Google Colab) ---
try:
    from google.colab import drive
    drive.mount('/content/drive')
    print("Google Drive mounted successfully.")
    # Define base path for data on Google Drive
    google_drive_base_path = '/content/drive/MyDrive/' # Adjust if your "MyDrive" has a different casing or path
except ImportError:
    print("Not running in Google Colab or google.colab.drive module not found. Assuming local file system.")
    google_drive_base_path = "" # Or set to your local base path if not in Colab

# Initialize SparkSession
spark = SparkSession.builder.appName("InterTradeTimeAnalysis").getOrCreate()

# Define the path to your trade data file
# Path now uses the mounted Google Drive path
input_base_dir_drive = os.path.join(google_drive_base_path, 'Tables/') # Path within MyDrive
trade_data_filename = "trade_data.txt" # Assuming this is the file name
trade_data_path = os.path.join(input_base_dir_drive, trade_data_filename)

print(f"Attempting to load trade data from: {trade_data_path}")

# --- 1. Load Trade Data ---
# Header from your file: CLIENTCODE,TRADE_DATE,TOTAL_GROSS_BROKERAGE_DAY
# Delimiter is likely comma (,) if it's a standard CSV, but your example shows values separated by comma
# Check if your TXT file is actually comma-separated or if the sample was just showing values.
# If it's still tilde (~), keep the delimiter as "~". If it's comma, change to ",".
# For now, I'll assume it's comma-separated based on the sample, but please verify.

try:
    trade_df = spark.read.format("csv") \
        .option("header", "true") \
        .option("delimiter", ",") \
        .option("inferSchema", "false") \
        .load(trade_data_path)

    # Select and cast necessary columns, using exact names from your file
    trade_df = trade_df.select(
        col("CLIENTCODE").alias("ClientCode"), # Using an alias for consistency if preferred, or just use "CLIENTCODE"
        to_date(col("TRADE_DATE"), "dd/MM/yyyy").alias("TradeDate") # Corrected column name and date format
        # col("TOTAL_GROSS_BROKERAGE_DAY") # Not needed for this calculation
    )

    # Filter out any rows where TradeDate couldn't be parsed (became null)
    trade_df = trade_df.filter(col("TradeDate").isNotNull())

    print("Trade data loaded and dates parsed successfully.")
    trade_df.show(5, truncate=False)
    print(f"Total trade records after date parsing: {trade_df.count()}")

except Exception as e:
    print(f"Error loading or parsing trade data: {e}")
    spark.stop()
    exit()


# --- 2. Prepare Data: Get Distinct Trading Days per Client ---
distinct_trade_days_df = trade_df.select("ClientCode", "TradeDate").distinct()
print(f"Total distinct ClientCode-TradeDate pairs: {distinct_trade_days_df.count()}")


# --- 3. Calculate Previous Trade Date for Each Client ---
window_spec = Window.partitionBy("ClientCode").orderBy("TradeDate")
trades_with_prev_date_df = distinct_trade_days_df.withColumn("Previous_TradeDate", lag("TradeDate", 1).over(window_spec))

print("Calculated previous trade date (sample):")
trades_with_prev_date_df.orderBy("ClientCode", "TradeDate").show(10, truncate=False)

# --- 4. Calculate Days Between Consecutive Trades ---
inter_trade_times_df = trades_with_prev_date_df.withColumn(
    "Days_Between_Trades",
    datediff(col("TradeDate"), col("Previous_TradeDate"))
)
```

```
print("Calculated days between trades (sample):")
inter_trade_times_df.orderBy("ClientCode", "TradeDate").show(10, truncate=False)

# --- 5. Filter out Nulls and Calculate Overall Average ---
valid_inter_trade_times_df = inter_trade_times_df.filter(col("Days_Between_Trades").isNotNull())

# valid_inter_trade_times_df.persist() # Optional

if valid_inter_trade_times_df.count() == 0:
    print("No valid inter-trade time periods found.")
    average_days_between_trades = None
    total_periods_calculated = 0
else:
    result_df = valid_inter_trade_times_df.agg(
        avg("Days_Between_Trades").alias("Average_Days_Between_Trades"),
        count("*").alias("Total_Inter_Trade_Periods")
    )

    print("Overall statistics for days between trades:")
    result_df.show(truncate=False)

    average_days_between_trades_row = result_df.first()
    if average_days_between_trades_row:
        average_days_between_trades = average_days_between_trades_row["Average_Days_Between_Trades"]
        total_periods_calculated = average_days_between_trades_row["Total_Inter_Trade_Periods"]
        print(f"\nSingle Overall Average Time Between Trades: {average_days_between_trades:.2f} days")
        print(f"Based on {total_periods_calculated} inter-trade periods.")
    else:
        print("Could not retrieve average days between trades.")
        average_days_between_trades = None
        total_periods_calculated = 0

# --- (Optional) Further Analysis: Median and Distribution ---
if valid_inter_trade_times_df.count() > 0:
    quantiles = valid_inter_trade_times_df.approxQuantile("Days_Between_Trades", [0.25, 0.5, 0.75, 0.90, 0.95], 0.01)
    if quantiles:
        print("\nApproximate Quantiles for Days Between Trades:")
        print(f"  25th Percentile (Q1): {quantiles[0]:.2f} days")
        print(f"  50th Percentile (Median): {quantiles[1]:.2f} days")
        print(f"  75th Percentile (Q3): {quantiles[2]:.2f} days")
        print(f"  90th Percentile: {quantiles[3]:.2f} days")
        print(f"  95th Percentile: {quantiles[4]:.2f} days")

# if 'valid_inter_trade_times_df' in locals() and valid_inter_trade_times_df.is_cached:
#     valid_inter_trade_times_df.unpersist()

# Stop the SparkSession
spark.stop()
```

```
Mounted at /content/drive
Google Drive mounted successfully.
Attempting to load trade data from: /content/drive/MyDrive/Tables/trade_data.txt
Trade data loaded and dates parsed successfully.
+----------+----------+
|ClientCode|TradeDate |
+----------+----------+
|SD9627    |2020-08-04|
|AV3818    |2020-08-04|
|NR2513    |2020-08-04|
|BN1496    |2020-08-04|
|UM1246    |2020-08-04|
+----------+----------+
only showing top 5 rows

Total trade records after date parsing: 17254800
Total distinct ClientCode-TradeDate pairs: 17254800
Calculated previous trade date (sample):
+----------+----------+------------------+
|ClientCode|TradeDate |Previous_TradeDate|
+----------+----------+------------------+
|06537     |2021-07-22|NULL              |
|06537     |2022-07-14|2021-07-22        |
|1001AUCTC |2020-08-04|NULL              |
|1001AUCTC |2020-08-05|2020-08-04        |
|1001AUCTC |2020-08-06|2020-08-05        |
|1001AUCTC |2020-08-07|2020-08-06        |
|1001AUCTC |2020-08-10|2020-08-07        |
|1001AUCTC |2020-08-12|2020-08-10        |
|1001AUCTC |2020-08-13|2020-08-12        |
|1001AUCTC |2020-08-14|2020-08-13        |
+----------+----------+------------------+
only showing top 10 rows

Calculated days between trades (sample):
+----------+----------+------------------+-------------------+
```

```
|ClientCode|TradeDate |Previous_TradeDate|Days_Between_Trades|
+----------+----------+------------------+-------------------+
|06537     |2021-07-22|NULL              |NULL               |
|06537     |2022-07-14|2021-07-22        |357                |
|1001AUCTC |2020-08-04|NULL              |NULL               |
|1001AUCTC |2020-08-05|2020-08-04        |1                  |
|1001AUCTC |2020-08-06|2020-08-05        |1                  |
|1001AUCTC |2020-08-07|2020-08-06        |1                  |
|1001AUCTC |2020-08-10|2020-08-07        |3                  |
|1001AUCTC |2020-08-12|2020-08-10        |2                  |
|1001AUCTC |2020-08-13|2020-08-12        |1                  |
|1001AUCTC |2020-08-14|2020-08-13        |1                  |
+----------+----------+------------------+-------------------+
only showing top 10 rows

Overall statistics for days between trades:
+--------------------------+-------------------------+
|Average_Days_Between_Trades|Total_Inter_Trade_Periods|
+--------------------------+-------------------------+
|8.138317632354466         |16988962                 |
+--------------------------+-------------------------+
```

Start coding or generate with AI.