Lab 5 — Handling Dates & Multiple Date Formats

This lab helps learners understand how to clean and convert dates, especially when the dataset contains inconsistent date formats. This is a common real-world problem and an essential part of ETL pipelines.

Learning Objective

Learners will clean and standardize date columns that appear in multiple formats.

Learning Outcomes

- Identify inconsistent date formats
- Use `try_to_date()` for tolerant parsing
- Apply multiple fallbacks using `coalesce`
- Extract year, month, and day
- Write cleaned output to Delta

Dataset

Use this new file:

```
/Volumes/workspace/default/test/customer_dates_raw.csv
```

Sample fields:

- customer_id
- name
- signup_date (multiple formats)

Trainer will generate this dataset separately.

Step-by-Step Lab Instructions

Step 1 — Read the Dataset

```
path = '/Volumes/workspace/default/test/customer_dates_raw.csv'
df = spark.read.option('header','true').option('inferSchema','false').csv(path)
```

```
display(df)
df.printSchema()
```

Step 2 — Apply Multi-Format Date Parsing

```
from pyspark.sql import functions as F
from pyspark.sql.functions import try_to_date

cleaned = df.withColumn(
    'signup_date_clean',
    F.coalesce(
        try_to_date('signup_date', 'yyyy-MM-dd'),
        try_to_date('signup_date', 'MM/dd/yyyy'),
        try_to_date('signup_date', 'dd-MM-yyyy')
    )
)

display(cleaned)
```

Explain that `coalesce` picks the first non-null parsed value.

Step 3 — Extract Date Components

```
cleaned = cleaned.withColumn('year', F.year('signup_date_clean'))
                 .withColumn('month', F.month('signup_date_clean'))
                 .withColumn('day', F.dayofmonth('signup_date_clean'))

display(cleaned)
```

Step 4 — Identify Rows That Failed Parsing

```
failed = cleaned.filter(F.col('signup_date_clean').isNull())
display(failed)
```

Trainer: Explain why some rows fail—typos, invalid patterns, or empty values.

Step 5 — Write Output to Silver Layer

```
silver_path = '/Volumes/workspace/default/test/customer_dates_silver'
cleaned.write.format('delta').mode('overwrite').option('overwriteSchema','true').save(silver_path
```

Post-Lab Practice

1. Add a column that identifies weekend signups.
2. Convert signup date to a string in `yyyyMMdd` format.
3. Create a new column showing quarter (Q1–Q4).

End of Lab 5 — Handling Dates