databricks 01.2 - Delta Live Tables - Python

(https://databricks.com)

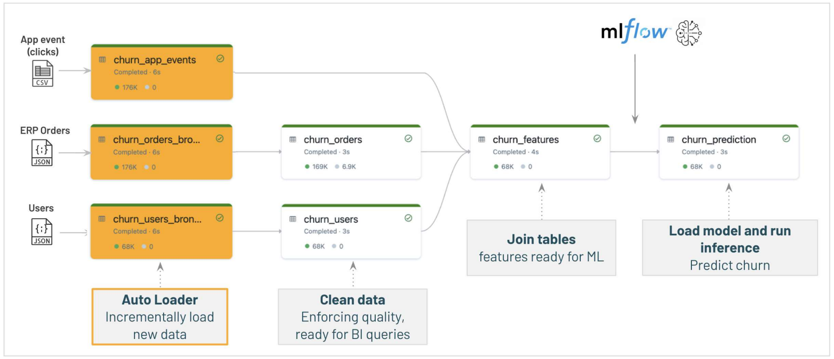
```
# The required imports that define the @dlt decorator
import dlt
from pyspark.sql import functions as F

# The path to the blob storage with the raw data
rawDataDirectory = "/cloud_lakehouse_labs/retail/raw"
eventsRawDataDir = rawDataDirectory + "/events"
ordersRawDataDir = rawDataDirectory + "/orders"
usersRawDataDir = rawDataDirectory + "/users"
```

1/ Loading our data using Databricks Autoloader (cloud_files)

Autoloader allow us to efficiently ingest millions of files from a cloud storage, and support efficient schema inference and evolution at scale.

Let's use it to our pipeline and ingest the raw JSON & CSV data being delivered in our blob cloud storage.



Ingest raw app events stream in incremental mode

```
@dlt.create_table(comment="Application events and sessions")
@dlt.expect("App events correct schema", "_rescued_data IS NULL")
def churn_app_events():
  return (
    spark.readStream.format("cloudFiles")
      .option("cloudFiles.format", "csv")
      .option("cloudFiles.inferColumnTypes", "true")
      .load(eventsRawDataDir))
```

Ingest raw orders from ERP

```
@dlt.create_table(comment="Spending score from raw data")
@dlt.expect("Orders correct schema", "_rescued_data IS NULL")
def churn_orders_bronze():
  return (
    spark.readStream.format("cloudFiles")
      .option("cloudFiles.format", "json")
      .option("cloudFiles.inferColumnTypes", "true")
      .load(ordersRawDataDir))
```

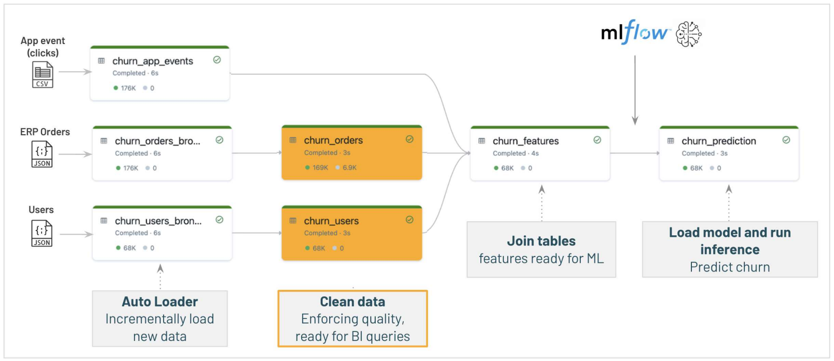
Ingest raw user data

```
@dlt.create_table(comment="Raw user data coming from json files ingested in incremental with Auto Loader to support schema inference and
evolution")
@dlt.expect("Users correct schema", "_rescued_data IS NULL")
def churn_users_bronze():
  return (
    spark.readStream.format("cloudFiles")
      .option("cloudFiles.format", "json")
      .option("cloudFiles.inferColumnTypes", "true")
      .load(usersRawDataDir))
```

2/ Enforce quality and materialize our tables for Data Analysts

The next layer often call silver is consuming **incremental** data from the bronze one, and cleaning up some information.

We're also adding an expectation (https://docs.databricks.com/workflows/delta-live-tables/delta-live-tables-expectations.html) on different field to enforce and track our Data Quality. This will ensure that our dashboard are relevant and easily spot potential errors due to data anomaly.



Clean and anonymise User data

```
@dlt.create_table(comment="User data cleaned and anonymized for analysis.")
@dlt.expect_or_drop("user_valid_id", "user_id IS NOT NULL")
def churn_users():
  return (dlt
          .read_stream("churn_users_bronze")
          .select(F.col("id").alias("user_id"),
                  F.sha1(F.col("email")).alias("email"),
                  F.to_timestamp(F.col("creation_date"), "MM-dd-yyyy HH:mm:ss").alias("creation_date"),
                  F.to_timestamp(F.col("last_activity_date"), "MM-dd-yyyy HH:mm:ss").alias("last_activity_date"),
                  F.initcap(F.col("firstname")).alias("firstname"),
                  F.initcap(F.col("lastname")).alias("lastname"),
                  F.col("address"),
                  F.col("channel"),
                  F.col("country"),
                  F.col("gender").cast("int").alias("gender"),
                  F.col("age_group").cast("int").alias("age_group"),
                  F.col("churn").cast("int").alias("churn")))
```

Clean orders

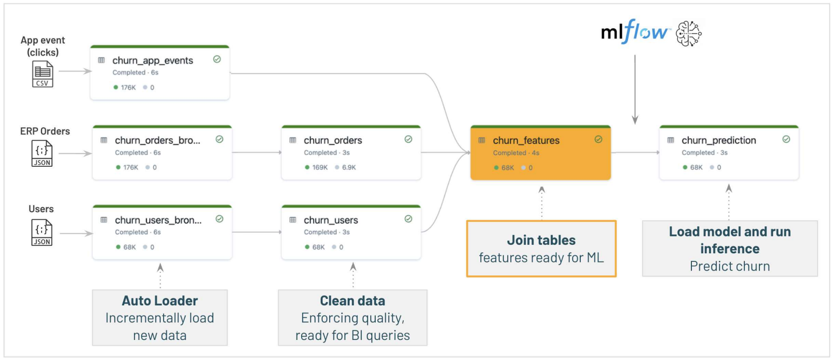
```
@dlt.create_table(comment="Order data cleaned and anonymized for analysis.")
@dlt.expect_or_drop("order_valid_id", "order_id IS NOT NULL")
@dlt.expect_or_drop("order_valid_user_id", "user_id IS NOT NULL")
def churn_orders():
  return (dlt
          .read_stream("churn_orders_bronze")
          .select(F.col("amount").cast("int").alias("amount"),
                  F.col("id").alias("order_id"),
                  F.col("user_id"),
                  F.col("item_count").cast("int").alias("item_count"),
                  F.to_timestamp(F.col("transaction_date"), "MM-dd-yyyy HH:mm:ss").alias("creation_date"))
         )
```

3/ Aggregate and join data to create our ML features

We're now ready to create the features required for our Churn prediction.

We need to enrich our user dataset with extra information which our model will use to help predicting churn, sucj as:

- last command date
- number of item bought
- number of actions in our website
- device used (ios/iphone)
- ...



Create the feature table