Airbnb Data Processing Pipeline (Batch Processing Triggered Every 30 Minutes)

This project implements a scalable and automated data pipeline for Airbnb, utilizing Azure services like Data Factory, Synapse Analytics, and Cosmos DB. It processes customer and booking data, performing data validation, transformation, and aggregation to provide enriched insights. The pipeline is designed for periodic updates and seamless integration, ensuring high data integrity and efficient analytics.

1. Pipeline Workflow

1.1 Data Sources

- Customer Data: Stored in ADLS under airbnbfile/customer_rawdata.
- Booking Data: Stored in Cosmos DB in the Airbnb.bookings container.

1.2 ADF Setup

1.2.1 Linked Services

- Synapse Analytics (airbnb.sqlpool)
- Cosmos DB (Airbnb.bookings)
- ADLS (airbnb storage account)

1.2.2 Datasets in ADF

1. bookingdatacosmos: Links to Cosmos DB booking data.
2. bookingfactsynapse: Links to Synapse bookings fact table.
3. customer_rawdata: Links to raw customer data in ADLS.
4. customer_dimsynapse: Links to Synapse customer dimension table.
5. customer_archive: Stores processed customer data in ADLS as CSV.

1.3 Pipeline 1: LoadCustomerDim

- Processes customer data from ADLS into customer_dim in Synapse.
- Activities:
  - Get Metadata Activity: Fetches file names from customer_rawdata.
  - For Each Activity: Iterates through file list.

- Copy Data Activity 1: Loads data from customer_rawdata to customer_dim (Upsert on customer_id).
- Copy Data Activity 2: Archives processed data in customer_archive.

1.4 Dataflow: DataflowTransform

- Processes booking data from Cosmos DB into Synapse.
- Steps:
  - Source: Reads bookingdatacosmos dataset.
  - Conditional Split: Filters valid and invalid bookings.
  - Derived Column: Adds computed attributes (16 columns).
  - Lookup Activity: Matches bookings with bookings_fact.
  - Alter Row Activity: Defines insert/update conditions.
  - Sink: Loads validated data into bookings_fact (Upsert logic).

1.5 Pipeline 2: LoadBookingFacts

- Triggers DataflowTransform and a stored procedure.
- Activities:
  - Dataflow Activity: Runs DataflowTransform.
  - Stored Procedure Activity: Calls airbnb.BookingAggregation.

1.6 Pipeline 3: AirbnbCDCPipeline

- Orchestrates LoadCustomerDim and LoadBookingFacts.
- Activities:
  - Executes LoadCustomerDim.
  - Executes LoadBookingFacts.

2. SQL Components

2.1 Tables in Synapse

- customer_dim: Enriched customer data (total bookings, total spent, preferred language).

- bookings_fact: Stores booking details (stay duration, booking year/month, amounts).

2.2 Aggregation Table

- BookingCustomerAggregation: Summarizes bookings by country (total bookings, amount, last booking date).

2.3 Stored Procedure: BookingAggregation

- Refreshes BookingCustomerAggregation by aggregating customer_dim and bookings_fact.

3. Python Integration with Cosmos DB

- Generates and inserts mock booking data.
- Key Features:
  - Uses Faker to generate realistic records.
  - Inserts data into Cosmos DB bookings container.
  - Simulates real-time ingestion for pipeline testing.

4. Key Features and Highlights

- Data Validation: Ensures valid records through conditional split.
- Upsert Logic: Maintains data integrity in customer_dim and bookings_fact.
- Automated Aggregation: BookingAggregation ensures updated metrics.
- Archival Mechanism: Stores processed customer data in ADLS.
- Modular Design: Pipelines operate independently or as part of AirbnbCDCPipeline.

5. Conclusion

This project showcases a scalable and efficient batch data pipeline for Airbnb, leveraging Azure services for data validation, transformation, and aggregation. It ensures high accuracy, modularity, and automation, making it ideal for periodic updates and analytics.