Capstone Project

Project Title
Advanced ETL Pipeline for Real-Time Fraud Detection and Customer 360 in Global Banking

Background
GlobalBank, a multinational financial institution operating in 50 countries, is facing increasing challenges in fraud detection and customer relationship management. The bank processes millions of transactions daily across various channels (online, mobile, ATM, in-branch) and needs to enhance its ability to detect fraudulent activities in real-time while also improving its customer service through a unified customer view.

Problem Statement
GlobalBank needs to develop an advanced, scalable, and real-time data pipeline that can ingest, process, and analyse large volumes of diverse data to achieve two primary objectives:

1. Implement a real-time fraud detection system capable of identifying potentially fraudulent transactions across all channels with high accuracy and low latency.
2. Create a comprehensive Customer 360 view by integrating data from multiple sources, enabling personalized customer service and targeted marketing initiatives.

The solution must be built using modern data engineering tools and practices, ensuring high performance, scalability, data security, and compliance with international banking regulations.

Project Requirements

1. Data Ingestion and Storage:
    Design and implement a system to ingest data from multiple sources, including real-time transaction streams, batch updates of customer profiles, and external data sources like credit bureaus and watchlists.
    Utilize AWS S3 as a raw data landing zone and Snowflake as the core data warehouse.
    Implement Snowpipe for continuous loading of data from S3 to Snowflake.

2. Data Transformation and Modeling:
    Utilize dbt (Data Build Tool) to create a modular and maintainable transformation layer.

Develop dbt models for data cleaning, normalization, and feature engineering for fraud detection and Customer 360 view.
Implement slowly changing dimensions (SCD) for customer attributes.
Create a star schema for analytical queries on customer data.

3. Real-Time Fraud Detection:
   Develop a streaming data pipeline using Snowflake streams and tasks.
   Implement both rule-based models for fraud detection.
   Create User-Defined Functions (UDFs) in Snowflake for fraud detection algorithms.
   Set up a system for periodic model retraining and rule updates.

4. Customer 360 View:
   Design and implement a unified customer profile by merging data from multiple sources.
   Create materialized views for frequently accessed customer data.
   Develop stored procedures for complex customer segmentation and analysis.

5. Pipeline Orchestration:
   Use Snowflake tasks and stored procedures to orchestrate the ETL pipeline.
   Design a workflow for data ingestion, transformation, fraud detection, and Customer 360 view updates.
   Implement proper task dependencies and parallel execution where possible.

6. Performance Optimization:
   Optimize query performance using appropriate clustering keys and search optimization for large tables.
   Implement materialised views for frequently accessed data.
   Develop a strategy for efficiently managing single-cluster warehouses:
      Configure warehouse auto-suspension and auto-resume for cost efficiency.
      Implement proper warehouse sizing based on workload requirements.
      Use resource monitors to track and control warehouse usage.

7. Security and Compliance:
   Implement row-level security and column-level encryption in Snowflake.
   Create secure views for data sharing across departments.

8. Monitoring and Alerting:
   Use Snowflake's native monitoring features to track pipeline status and performance.
   Set up email alerts for critical pipeline events and potential fraud detection.

9. Reporting and Visualization:

Utilize Power BI to create dashboards for fraud monitoring, Customer 360 view, and operational metrics of the data pipeline.

Connect Power BI to Snowflake for real-time data visualization.

Deliverables:

1. Fully functional ETL pipeline implemented in Airflow, dbt, and Snowflake.
2. Real-time fraud detection system integrated with the pipeline.
3. Customer 360 view with associated analytics capabilities.
5. Set of SQL scripts, dbt models, and Airflow DAGs used in the implementation.
6. QuickSight dashboards for monitoring and analysis.

Data Simulation and Schema:

1. Transaction Data Simulation: Create a Python script to generate transaction data. The script should:
    - Generate transactions for multiple channels (online, mobile, ATM, in-branch)
    - Vary transaction frequency based on time of day and day of week
    - Include a mix of transaction types (purchases, transfers, withdrawals, deposits)
    - Inject anomalies and potential fraud patterns at a controlled rate

Schema for transactions:

```
transaction_id STRING,
customer_id STRING,
transaction_date TIMESTAMP_NTZ,
amount FLOAT,
currency STRING,
transaction_type STRING,
channel STRING,
merchant_name STRING,
merchant_category STRING,
location_country STRING,
location_city STRING,
is_flagged BOOLEAN
```

2. Customer Data Simulation: Create a script to generate customer profiles with realistic distributions of:
    - Age, gender, occupation
    - Account types and balances
    - Customer tenure
    - Contact information

Schema for customer data:

```
customer_id STRING,
```

first_name STRING,
last_name STRING,
date_of_birth DATE,
gender STRING,
email STRING,
phone_number STRING,
address STRING,
city STRING,
country STRING,
occupation STRING,
income_bracket STRING,
customer_since DATE

3. Account Data Simulation: Generate account data linked to customers, including:
    - Multiple account types (checking, savings, credit card, loan)
    - Account balances and limits
    - Account status (active, dormant, closed)

Schema for account data:

account_id STRING,
customer_id STRING,
account_type STRING,
account_status STRING,
open_date DATE,
current_balance FLOAT,
currency STRING,
credit_limit FLOAT

4. External Data Simulation: Create simulated data for external sources:
    - Credit bureau data (credit scores, credit history)
    - Watchlists for anti-money laundering (AML) checks

Schema for credit data:

customer_id STRING,
credit_score INT,
number_of_credit_accounts INT,
total_credit_limit FLOAT,
total_credit_used FLOAT,
number_of_late_payments INT,

bankruptcies INT

Schema for watchlist:
entity_id STRING,
entity_name STRING,
entity_type STRING,
risk_category STRING,
listed_date DATE,
source STRING

5. Data Generation Process:
    Create a main Python script that calls individual data generation functions for each entity.
    Use libraries like Faker to generate realistic looking data.
    Ensure referential integrity across tables (e.g., transactions reference valid customer_ids and account_ids).
    Generate data for a specified time range (e.g., last 2 years of transactions).
    Output data in CSV format for easy ingestion into S3 and subsequently Snowflake.
6. Data Ingestion Simulation:
    Set up an AWS S3 bucket to receive the generated data.
    Create a script to periodically upload new transaction data to S3, simulating real-time data flow.
    For batch data (like customer profiles), upload updates daily or weekly.
7. Snowflake Setup:
    Create a staging area in Snowflake to receive data from S3.
    Set up Snowpipe to continuously ingest data from S3 into raw tables in Snowflake.
    Create views on top of raw tables to implement any initial data quality rules.