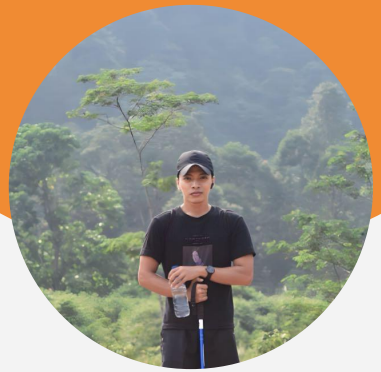
Spotify's End-to-End Insight ETL including replication with Debezium

Rahmadiyan Muhammad



Rahmadiyan M

Education

Bina Sarana Informatika - Sistem Informasi (ongoing)

Working

PT. Bank BTPN - Big Data Infrastructure & Operation Support

Project Background

An end-to-end solution for ingesting, processing, analyzing, and visualizing Spotify user listening data. It's goal is to provide comprehensive insight into user listening patterns and preferences by leveraging various data engineering and analytics techniques.

Project repo:
https://github.com/rahmadiyann/db_final_project

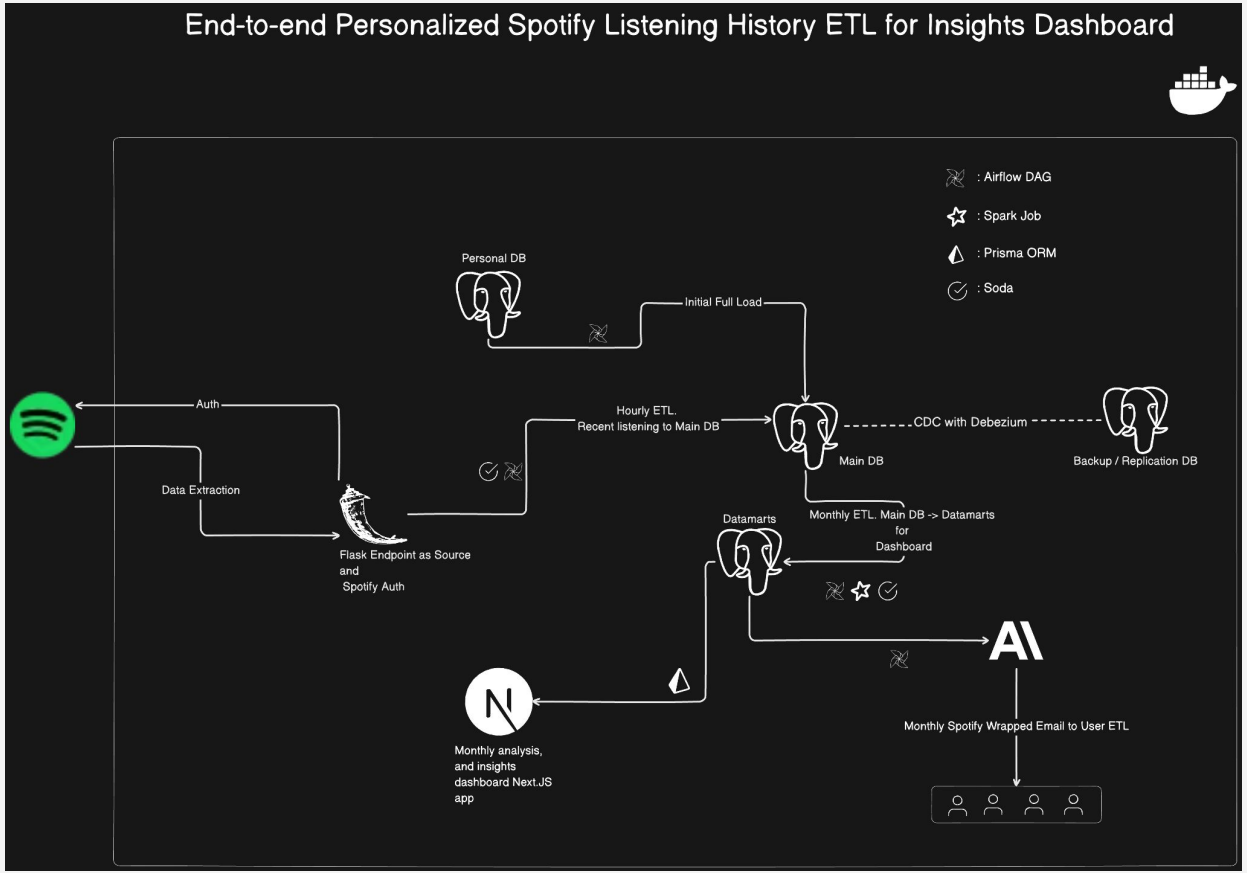
Problem Statement

We employ API integration, database management, data quality checks, data processing and transformation, as well as email notification and efficient data storage and management.

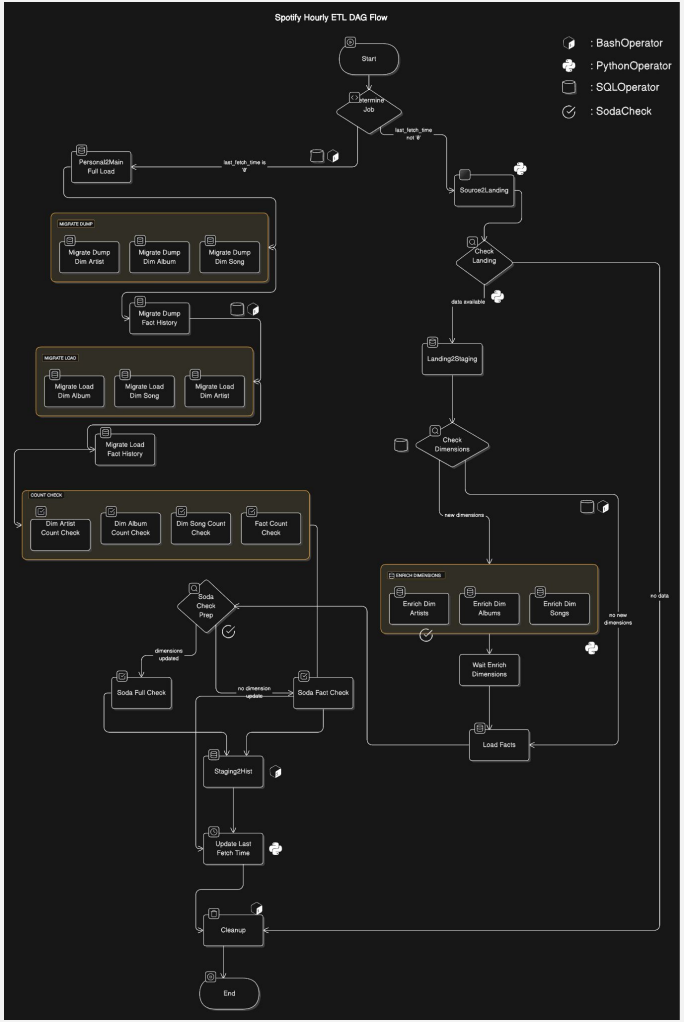
The success of the project can be measured such as data accuracy and completeness, data processing performance and the quality and relevance of generated insights and recommendation

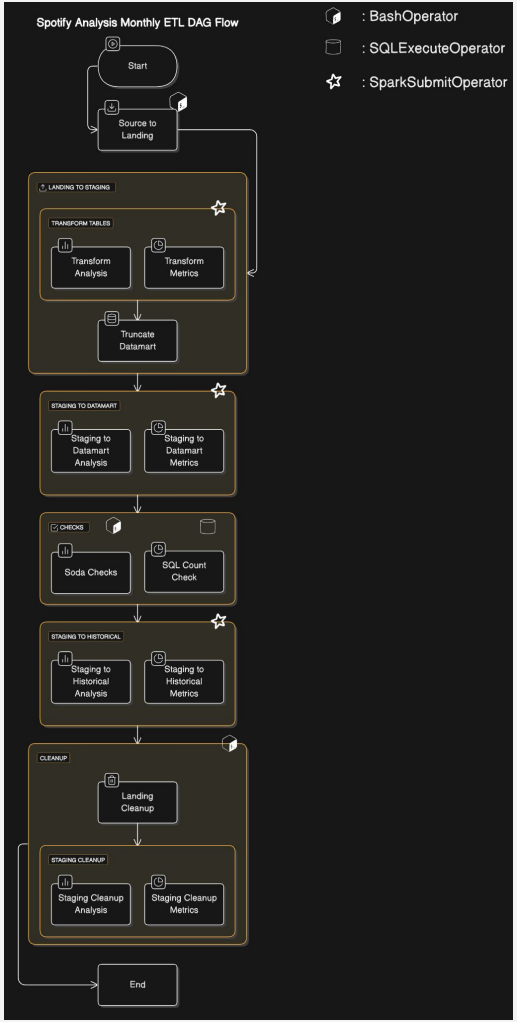
ibimbing

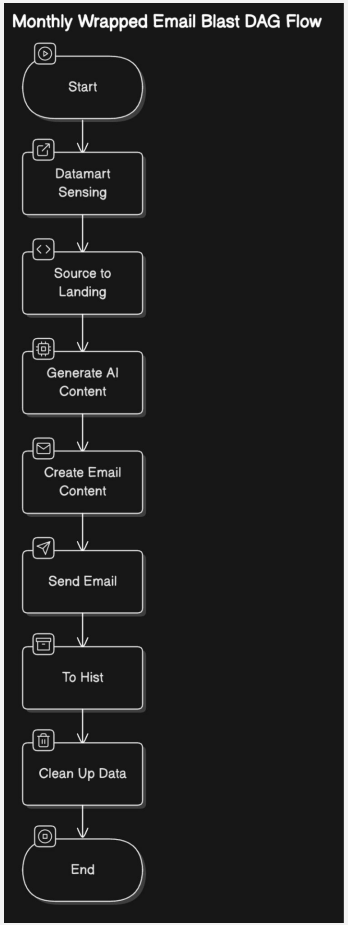
Data Platform Understanding


End-to-end Personalized Spotify Listening History ETL for Insights Dashboard
: Airflow DAG
: Spark Job
: Prisma ORM
: Soda
Personal DB
Initial Full Load
Auth
Hourly ETL.
Recent listening to Main DB
CDC with Debezium
Main DB
Backup / Replication DB
Data Extraction
Flask Endpoint as Source
and
Spotify Auth
Datamarts
Monthly ETL. Main DB -> Datamarts
for
Dashboard
Monthly analysis,
and insights
dashboard Next.JS
app
Monthly Spotify Wrapped Email to User ETL

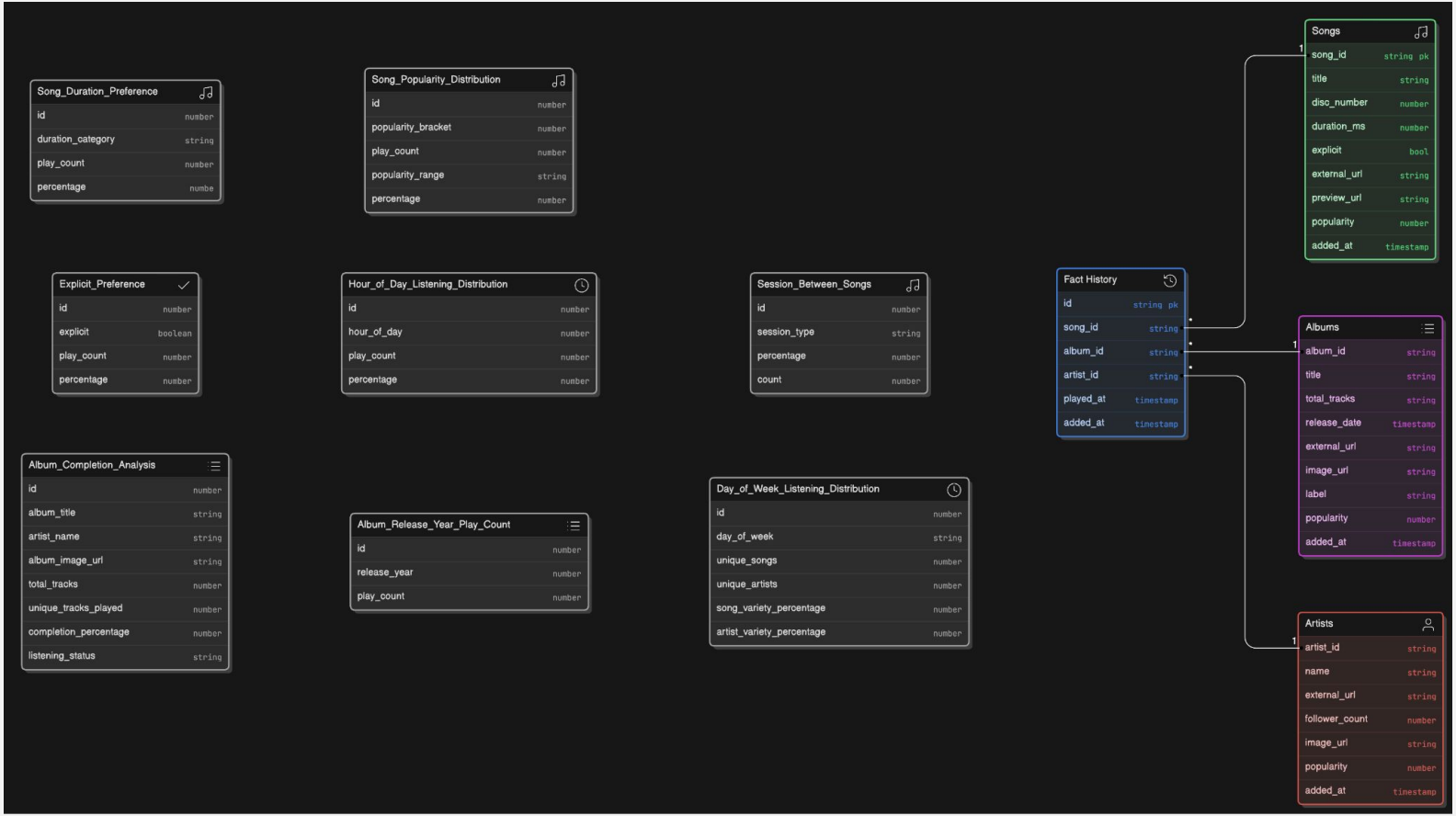
Transformation & Consideration


Spotify Hourly ETL DAG Flow
: BashOperator
: PythonOperator
: SQLOperator
: SodaCheck
Start
Determine Job
last_fetch_time not 'd'
last_fetch_time is 'd'
Personal2Main Full Load
Source2Landing
MIGRATE DUMP
Migrate Dump Dim Artist
Migrate Dump Dim Album
Migrate Dump Dim Song
Check Landing
data available
Migrate Dump Fact History
Landing2Staging
MIGRATE LOAD
Migrate Load Dim Album
Migrate Load Dim Song
Migrate Load Dim Artist
Check Dimensions
Migrate Load Fact History
new dimensions
COUNT CHECK
Dim Artist Count Check
Dim Album Count Check
Dim Song Count Check
Fact Count Check
ENRICH DIMENSIONS
Enrich Dim Artists
Enrich Dim Albums
Enrich Dim Songs
no new dimensions
no data
Soda Check Prep
dimensions updated
Soda Full Check
no dimension update
Soda Fact Check
Wait Enrich Dimensions
Load Facts
Staging2Hist
Update Last Fetch Time
Cleanup
End


Spotify Analysis Monthly ETL DAG Flow
: BashOperator
: SQLExecuteOperator
: SparkSubmitOperator
Start
Source to Landing
LANDING TO STAGING
TRANSFORM TABLES
Transform Analysis
Transform Metrics
Truncate Datamart
STAGING TO DATAMART
Staging to Datamart Analysis
Staging to Datamart Metrics
CHECKS
Soda Checks
SQL Count Check
STAGING TO HISTORICAL
Staging to Historical Analysis
Staging to Historical Metrics
CLEANUP
Landing Cleanup
STAGING CLEANUP
Staging Cleanup Analysis
Staging Cleanup Metrics
End


Monthly Wrapped Email Blast DAG Flow
Start
Datamart Sensing
Source to Landing
Generate AI Content
Create Email Content
Send Email
To Hist
Clean Up Data
End

Data Modeling (Business)


Song_Duration_Preference
id number
duration_category string
play_count number
percentage numbe
Song_Popularity_Distribution
id number
popularity_bracket number
play_count number
popularity_range string
percentage number
Explicit_Preference
id number
explicit boolean
play_count number
percentage number
Hour_of_Day_Listening_Distribution
id number
hour_of_day number
play_count number
percentage number
Session_Between_Songs
id number
session_type string
percentage number
count number
Fact History
id string pk
song_id string
album_id string
artist_id string
played_at timestamp
added_at timestamp
Songs
song_id string pk
title string
disc_number number
duration_ms number
explicit bool
external_url string
preview_url string
popularity number
added_at timestamp
Albums
album_id string
title string
total_tracks string
release_date timestamp
external_url string
image_url string
label string
popularity number
added_at timestamp
Artists
artist_id string
name string
external_url string
follower_count number
image_url string
popularity number
added_at timestamp
Album_Completion_Analysis
id number
album_title string
artist_name string
album_image_url string
total_tracks number
unique_tracks_played number
completion_percentage number
listening_status string
Album_Release_Year_Play_Count
id number
release_year number
play_count number
Day_of_Week_Listening_Distribution
id number
day_of_week string
unique_songs number
unique_artists number
song_variety_percentage number
artist_variety_percentage number

Conclusion & Recommendation

This platform successfully ingest, process, and visualize user's spotify listening data using relevant Data Engineering tools such as Spark, and Airflow as well as maintaining backup using Debezium and data quality check using Soda Core. The output is as expected in a beautiful manner that user can see through the Next.JS dashboard.

Thank you.