TO THE NEW™

DRIVING INNOVATION
DELIVERING EXCELLENCE

Employee Data Management Systems

Bootcamp Capstone Project

Presented by: Kanishka Sharma, Himanshu Kumar

Table of Contents:

- Introduction
- Objectives and Key Outcomes
- System Architecture & Data Flow
- Data Processing Implementation
- Reporting, Monitoring, and Alerts
- Conclusion

Introduction

In today's fast-paced organizations, efficiently managing employee data is key to operational excellence. This project introduces a robust data management platform built on AWS that handles real-time employee data using Kafka and PySpark.

The system brings together diverse datasets—like employee records, leave history, and communication logs—and transforms them into structured, enriched outputs using AWS Glue. It enables daily and monthly analytics to detect unusual leave patterns, communication misuse, and even automates HR insights like strike tracking and salary adjustments.

Objectives and Key Outcomes

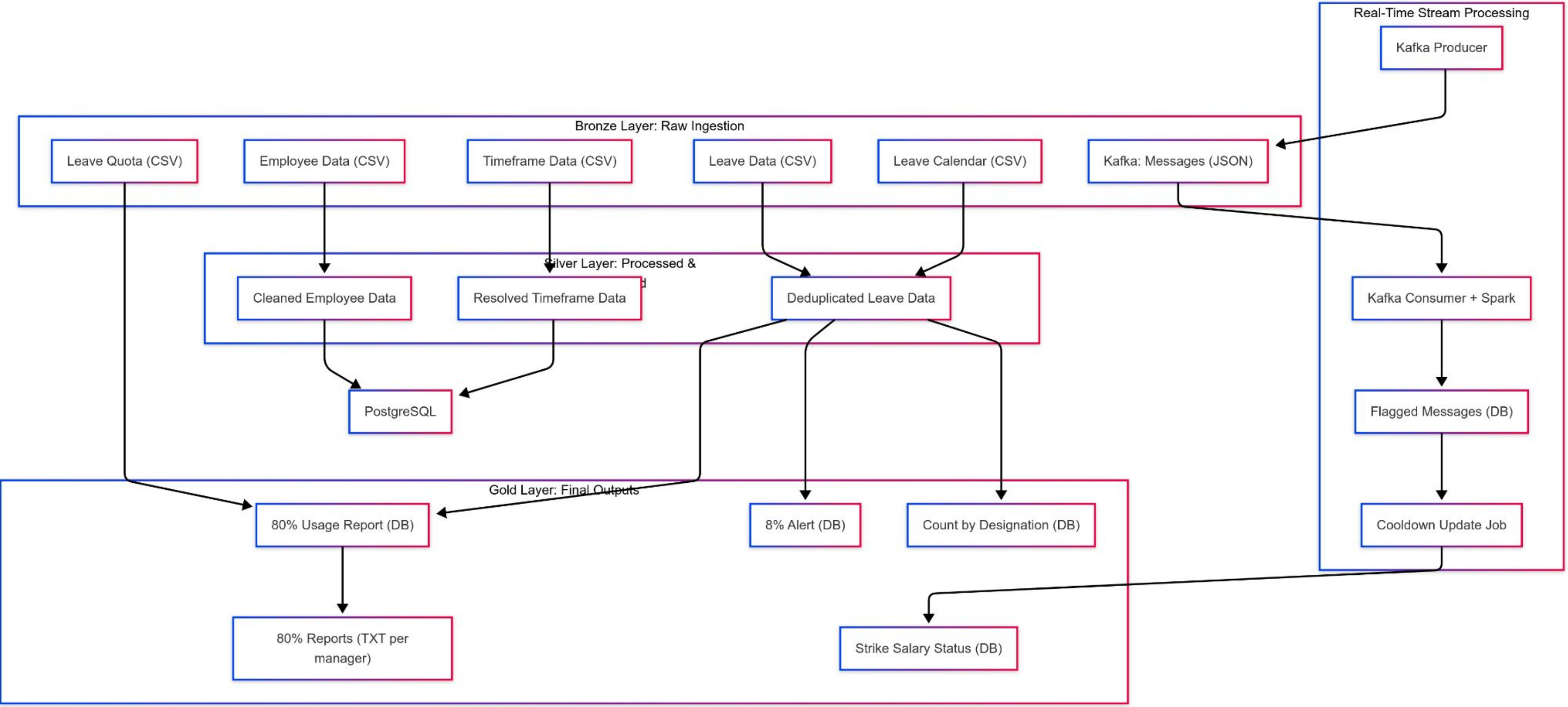
- Objectives:
  - Ingest and analyze employee data (leave, communication, records) in real time and batch modes.
  - Automate HR insights like leave abuse detection and strike tracking.
  - Ensure data reliability using scalable ETL pipelines and warehousing.

- Key Outcomes:
  - Built real-time data pipelines using Kafka, PySpark, and AWS Glue.
  - Detected leave misuse and flagged inappropriate communication.
  - Flexible: Supports batch and realtime data processing.
  - Generated automated HR reports with salary and strike enforcement logic.

System Architecture:

- Designed a scalable system to process employee data in real-time and batch modes.
- Ingested leave records, communication logs, and employee details via AWS and Kafka.
- Built automated ETL pipelines using PySpark and AWS Glue for reliable data transformation.
- Detected excessive leave usage through daily (8%) and monthly (80%) thresholds.
- Flagged inappropriate communication using keyword monitoring and Kafka streams.
- Implemented strike tracking, salary deductions, and manager-specific HR reports.

System Architecture:



Real-Time Stream Processing

Kafka Producer

Bronze Layer: Raw Ingestion

Leave Quota (CSV)

Employee Data (CSV)

Timeframe Data (CSV)

Leave Data (CSV)

Leave Calendar (CSV)

Kafka: Messages (JSON)

Silver Layer: Processed &

d

Cleaned Employee Data

Resolved Timeframe Data

Deduplicated Leave Data

Kafka Consumer + Spark

PostgreSQL

Flagged Messages (DB)

Gold Layer: Final Outputs

80% Usage Report (DB)

8% Alert (DB)

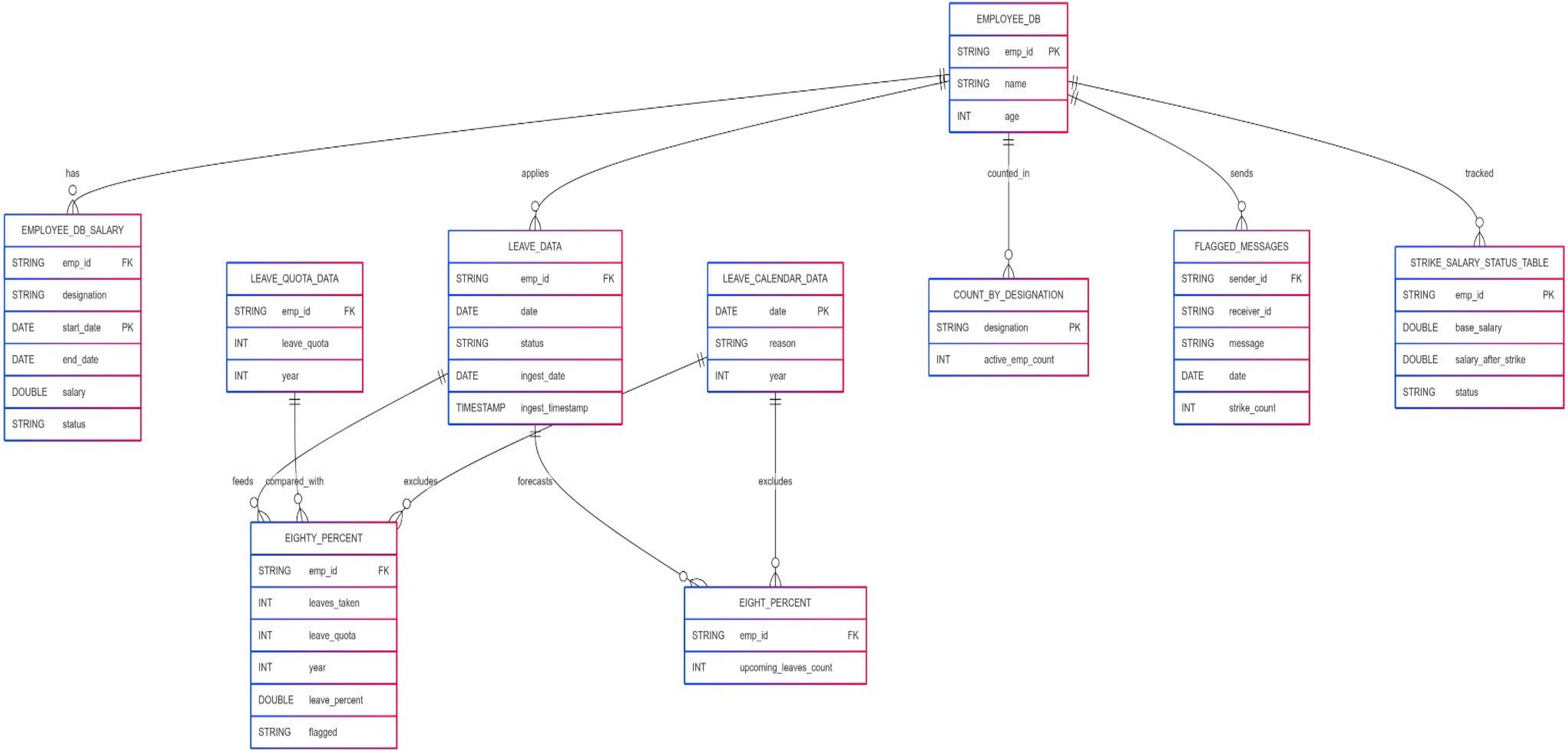
Count by Designation (DB)

Cooldown Update Job

80% Reports (TXT per manager)

Strike Salary Status (DB)

ER Diagram:



EMPLOYEE_DB

STRING	emp_id	PK
STRING	name	
INT	age	

EMPLOYEE_DB_SALARY

STRING	emp_id	FK
STRING	designation	
DATE	start_date	PK
DATE	end_date	
DOUBLE	salary	
STRING	status	

LEAVE_QUOTA_DATA

STRING	emp_id	FK
INT	leave_quota	
INT	year	

LEAVE_DATA

STRING	emp_id	FK
DATE	date	
STRING	status	
DATE	ingest_date	
TIMESTAMP	ingest_timestamp	

LEAVE_CALENDAR_DATA

DATE	date	PK
STRING	reason	
INT	year	

COUNT_BY_DESIGNATION

STRING	designation	PK
INT	active_emp_count	

FLAGGED_MESSAGES

STRING	sender_id	FK
STRING	receiver_id	
STRING	message	
DATE	date	
INT	strike_count	

STRIKE_SALARY_STATUS_TABLE

STRING	emp_id	PK
DOUBLE	base_salary	
DOUBLE	salary_after_strike	
STRING	status	

EIGHTY_PERCENT

STRING	emp_id	FK
INT	leaves_taken	
INT	leave_quota	
INT	year	
DOUBLE	leave_percent	
STRING	flagged	

EIGHT_PERCENT

STRING	emp_id	FK
INT	upcoming_leaves_count	

Data Flow:

- **Bronze Layer (Raw Ingestion):**
  Ingests CSVs from S3 (employee, leave, quota) and real-time Kafka messages; stores raw data in S3 for lineage.

- **Silver Layer (Processed Data):**
  Cleansed and enriched via AWS Glue & Spark; stored in PostgreSQL (e.g., leave data, flagged messages).

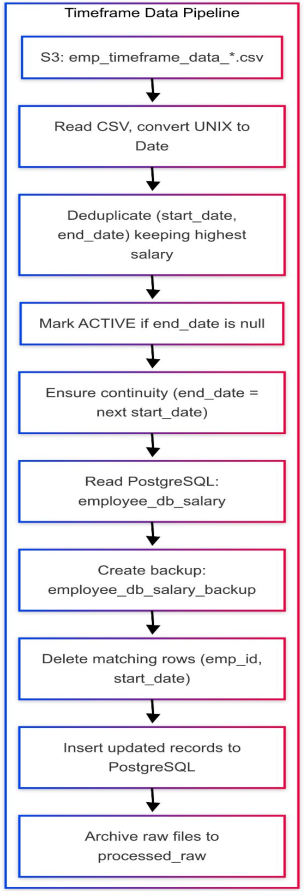
- **Gold Layer (Business Outputs):**
  Generates reports (e.g., leave thresholds, strike salary summaries); outputs stored in S3 & PostgreSQL.

- **Streaming + Batch Support:**
  Kafka for real-time messages, S3 for scheduled batch data ingestion.

- **Orchestration with Airflow:**
  Schedules daily/monthly Glue jobs with retry logic and checkpointing.

Employee Data and Timeframe Processing

- Ingests and cleans employee master and designation data daily.
- Maintains append-only tables with continuity logic and ACTIVE/INACTIVE status.
- Deduplicates based on timestamps and updates PostgreSQL incrementally.



Timeframe Data Pipeline

S3: emp_timeframe_data_*.csv

Read CSV, convert UNIX to Date

Deduplicate (start_date, end_date) keeping highest salary

Mark ACTIVE if end_date is null

Ensure continuity (end_date = next start_date)

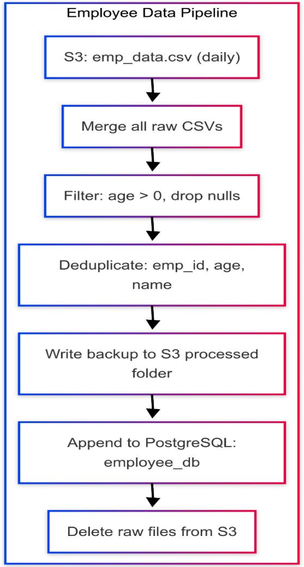
Read PostgreSQL: employee_db_salary

Create backup: employee_db_salary_backup

Delete matching rows (emp_id, start_date)

Insert updated records to PostgreSQL

Archive raw files to processed_raw



Employee Data Pipeline

S3: emp_data.csv (daily)

Merge all raw CSVs

Filter: age > 0, drop nulls

Deduplicate: emp_id, age, name

Write backup to S3 processed folder

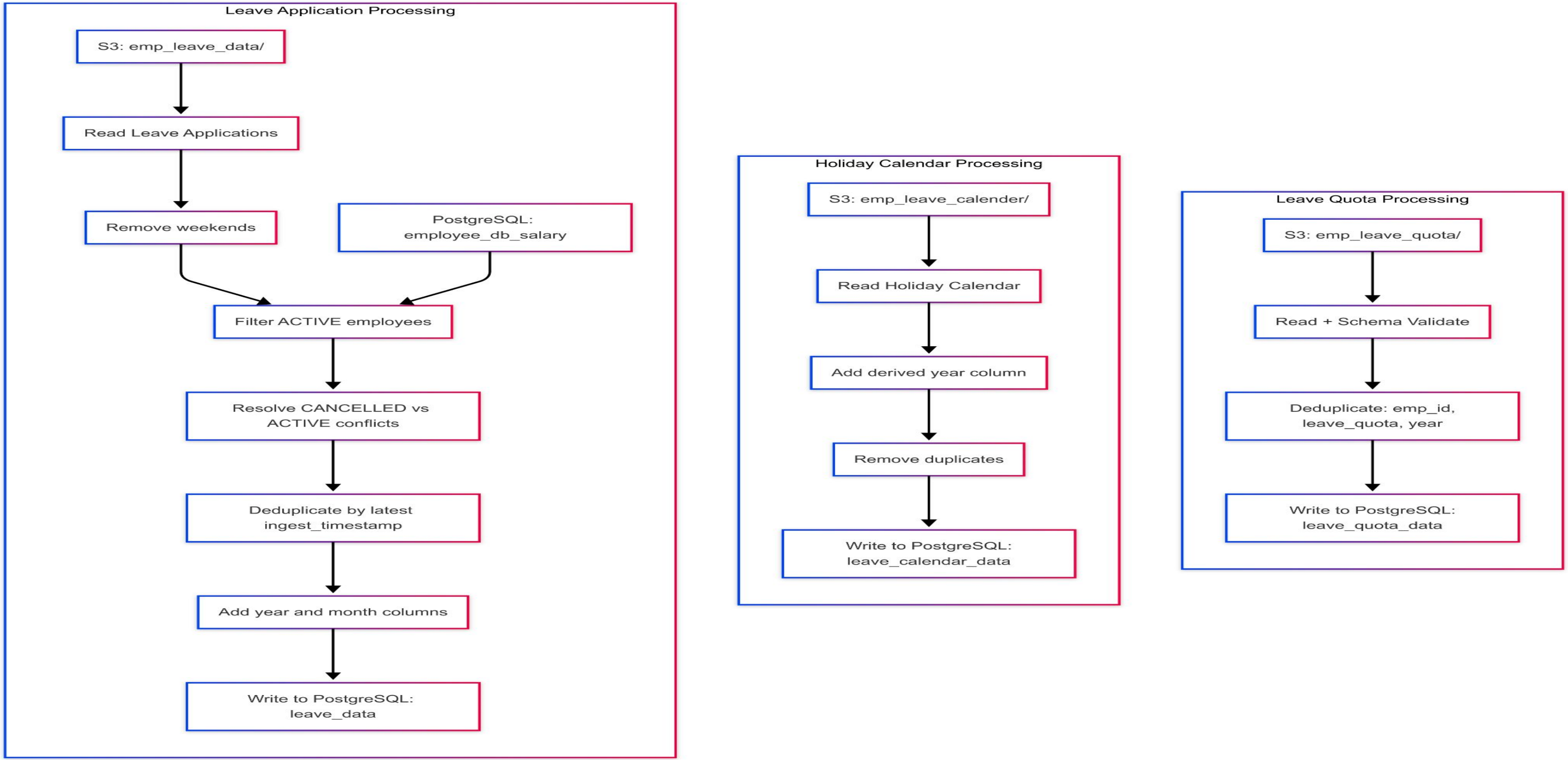
Append to PostgreSQL: employee_db

Delete raw files from S3

Employee Leave Data, Leave Quota & Calendar

- Processes leave quotas, public holidays, and daily leave applications.
- Stores cleaned data in PostgreSQL for dashboards and scheduled reports.
- Excludes holidays, weekends, and cancelled leaves from final tables, Deduplicated yearly quotas per employee.
- Cleaned public holidays for filtering leaves.
- Validated daily leave entries excluding weekends, holidays, and cancellations.

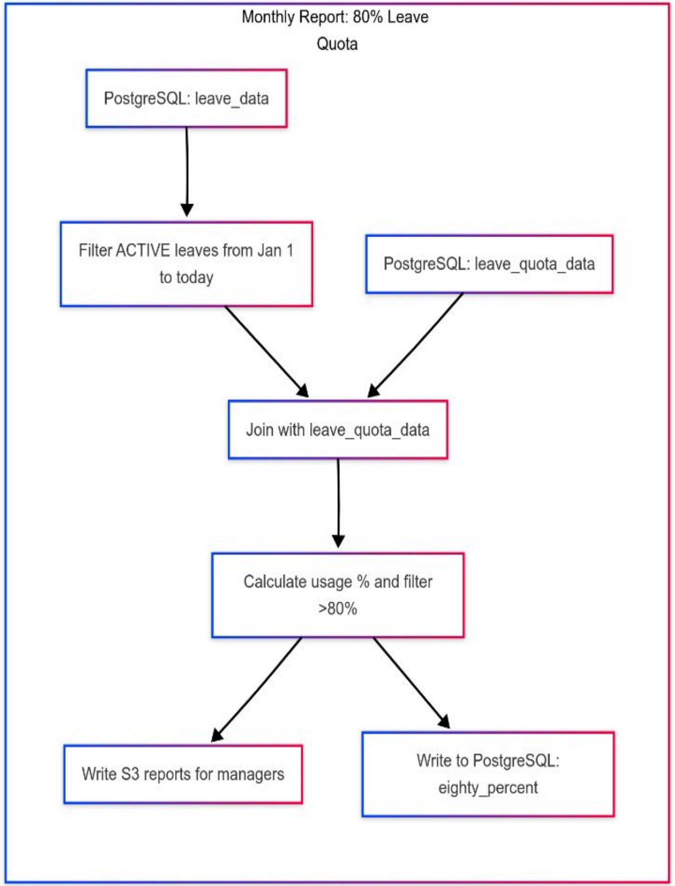
Employee Leave Data Processing

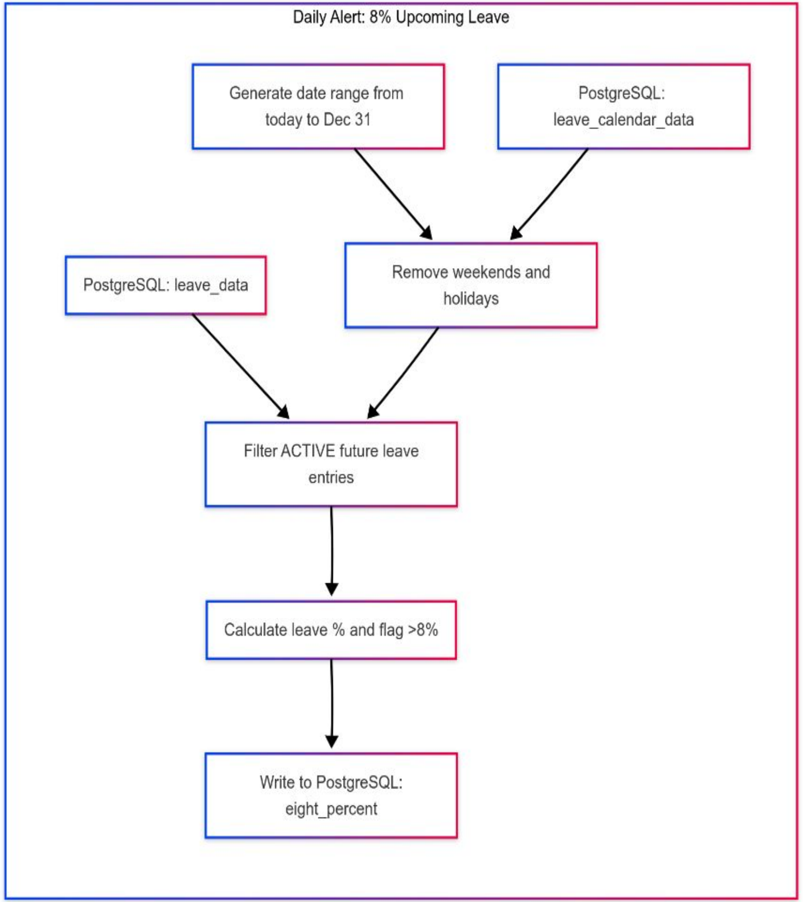

Leave Application Processing
S3: emp_leave_data/
Read Leave Applications
Remove weekends
PostgreSQL: employee_db_salary
Filter ACTIVE employees
Resolve CANCELLED vs ACTIVE conflicts
Deduplicate by latest ingest_timestamp
Add year and month columns
Write to PostgreSQL: leave_data
Holiday Calendar Processing
S3: emp_leave_calender/
Read Holiday Calendar
Add derived year column
Remove duplicates
Write to PostgreSQL: leave_calendar_data
Leave Quota Processing
S3: emp_leave_quota/
Read + Schema Validate
Deduplicate: emp_id, leave_quota, year
Write to PostgreSQL: leave_quota_data

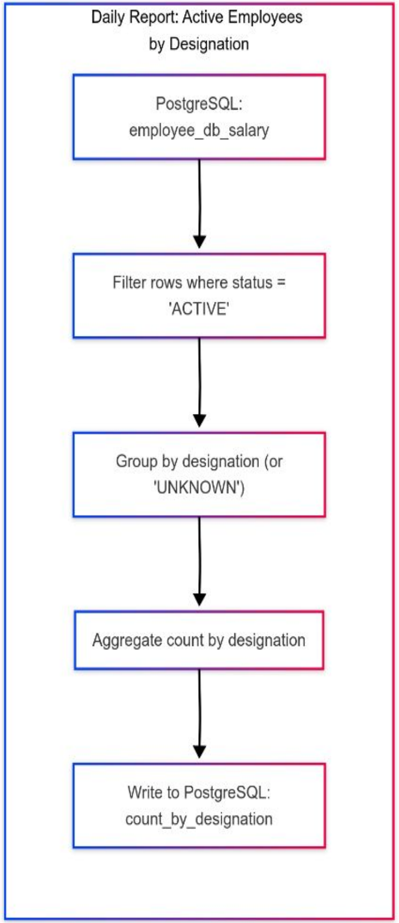
Employee Reporting and Alerting

- **Daily Report:** Tracks active employees by designation.
- **8% Alert:** Flags employees exceeding 8% of upcoming leave days.
- **80% Alert:** Flags those using over 80% of their annual leave quota; generates manager reports.

Employee Reporting and Alerting

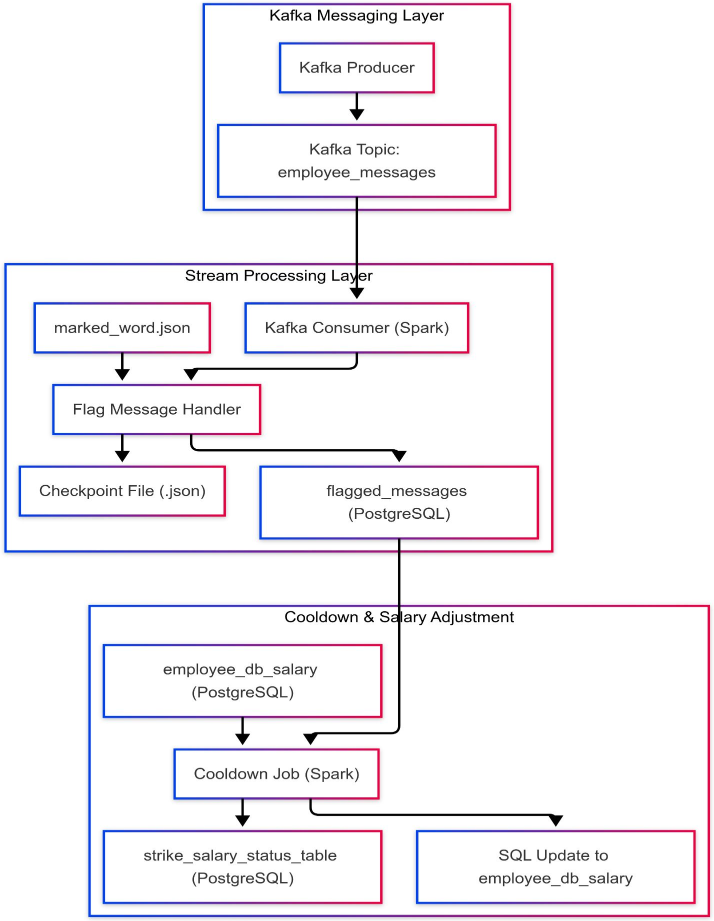

Monthly Report: 80% Leave Quota
PostgreSQL: leave_data
Filter ACTIVE leaves from Jan 1 to today
PostgreSQL: leave_quota_data
Join with leave_quota_data
Calculate usage % and filter >80%
Write S3 reports for managers
Write to PostgreSQL: eighty_percent


Daily Alert: 8% Upcoming Leave
Generate date range from today to Dec 31
PostgreSQL: leave_calendar_data
PostgreSQL: leave_data
Remove weekends and holidays
Filter ACTIVE future leave entries
Calculate leave % and flag >8%
Write to PostgreSQL: eight_percent


Daily Report: Active Employees by Designation
PostgreSQL: employee_db_salary
Filter rows where status = 'ACTIVE'
Group by designation (or 'UNKNOWN')
Aggregate count by designation
Write to PostgreSQL: count_by_designation

Real-Time Communication Monitoring

- Kafka-based pipeline processes real-time employee messages.
- Flags messages with reserved words and writes alerts to PostgreSQL.
- Maintains checkpoints for fault tolerance.
- Applies 10% salary deductions per strike with decay logic.
- Employees with ≥10 strikes marked INACTIVE and salary frozen.
- Updates PostgreSQL tables for strike status and employee state.


Kafka Messaging Layer
Kafka Producer
Kafka Topic: employee_messages
Stream Processing Layer
marked_word.json
Kafka Consumer (Spark)
Flag Message Handler
Checkpoint File (.json)
flagged_messages (PostgreSQL)
Cooldown & Salary Adjustment
employee_db_salary (PostgreSQL)
Cooldown Job (Spark)
strike_salary_status_table (PostgreSQL)
SQL Update to employee_db_salary

Airflow Orchestration:

1. **Start**: DAG begins with `start`, triggering `employee_data_job` and `employee_timeframe_job`.

2. **Parallel Branching**: These feed three paths — monthly check, designation report, and leave data processing.

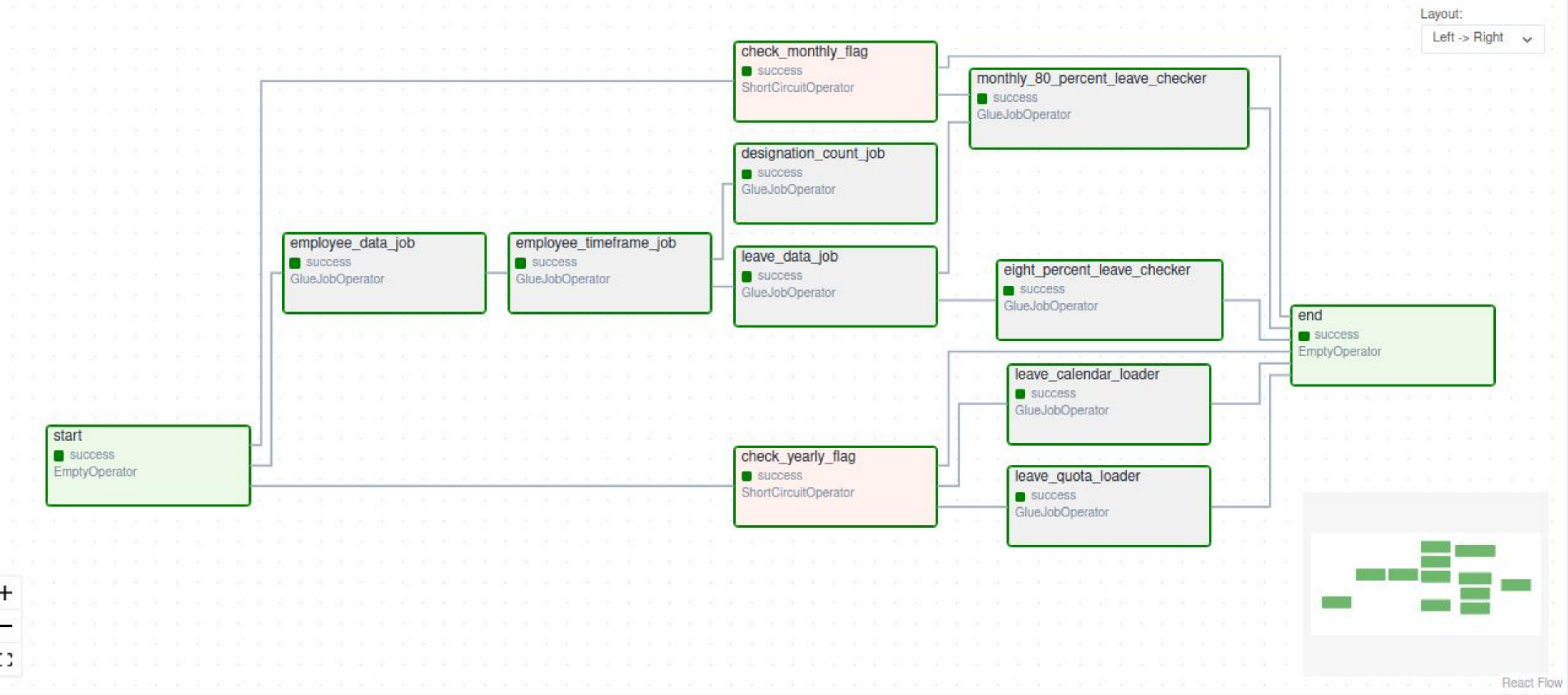
3. **Monthly Logic**: `check_monthly_flag` runs `monthly_80_percent_leave_checker` only on the 1st of each month.

4. **Daily Leave Checks**: `leave_data_job` processes leave entries and passes to `eight_percent_leave_checker`.

5. **Yearly Data Load**: `check_yearly_flag` ensures `leave_calendar_loader` and `leave_quota_loader` run only once a year.

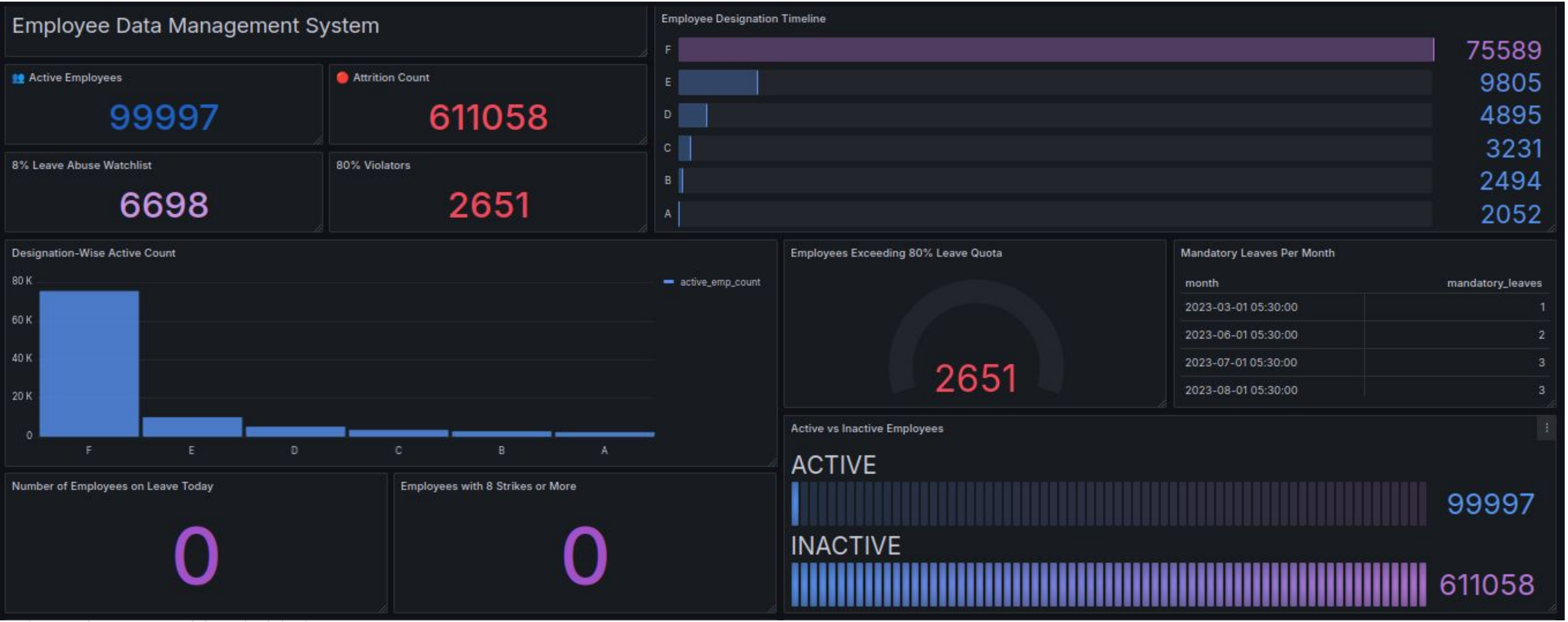
6. **End**: All paths merge at end, ensuring clean and complete execution flow.

Airflow:



Layout:
Left -> Right
check_monthly_flag
success
ShortCircuitOperator
monthly_80_percent_leave_checker
success
GlueJobOperator
designation_count_job
success
GlueJobOperator
employee_data_job
success
GlueJobOperator
employee_timeframe_job
success
GlueJobOperator
leave_data_job
success
GlueJobOperator
eight_percent_leave_checker
success
GlueJobOperator
end
success
EmptyOperator
leave_calendar_loader
success
GlueJobOperator
start
success
EmptyOperator
check_yearly_flag
success
ShortCircuitOperator
leave_quota_loader
success
GlueJobOperator
React Flow

Dashboard:



Employee Data Management System

Active Employees

99997

Attrition Count

611058

8% Leave Abuse Watchlist

6698

80% Violators

2651

Employee Designation Timeline

F	75589
E	9805
D	4895
C	3231
B	2494
A	2052

Designation-Wise Active Count

Employees Exceeding 80% Leave Quota

2651

Mandatory Leaves Per Month

month	mandatory_leaves
2023-03-01 05:30:00	1
2023-06-01 05:30:00	2
2023-07-01 05:30:00	3
2023-08-01 05:30:00	3

Number of Employees on Leave Today

0

Employees with 8 Strikes or More

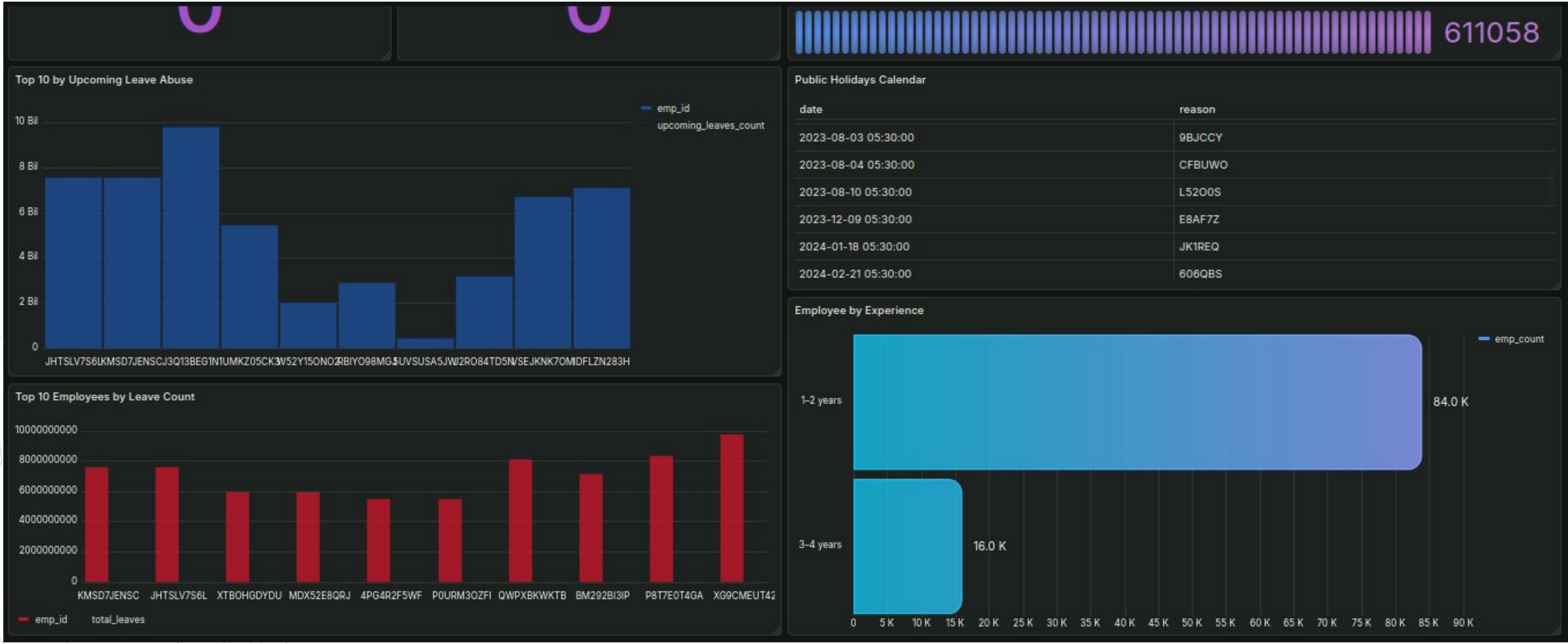
0

Active vs Inactive Employees

ACTIVE 99997

INACTIVE 611058

Dashboard:



Public Holidays Calendar

date	reason
2023-08-03 05:30:00	9BJCCY
2023-08-04 05:30:00	CFBUWO
2023-08-10 05:30:00	L52OOS
2023-12-09 05:30:00	E8AF7Z
2024-01-18 05:30:00	JK1REQ
2024-02-21 05:30:00	6O6QBS

Conclusion:

- The Employee Data Management System successfully demonstrates the integration of batch and streaming data pipelines to manage critical employee information at scale. By leveraging AWS services such as S3, Glue, and EC2-hosted PostgreSQL, along with Apache Kafka for real-time communication monitoring, the system ensures timely ingestion, accurate processing, and reliable reporting.

- This project showcases a scalable and fault-tolerant design aligned with modern data warehouse principles and lays the foundation for future enhancements such as alert automation, employee self-service dashboards, and advanced behavioral analytics.

Thank you!
ANY Questions?

TO THE NEW™

DRIVING INNOVATION DELIVERING EXCELLENCE

✉ info@tothenew.com

Scan to know more