**10Alytics Capstone Project**

**NYC Payroll Data Integration – Data Engineering Report**

**Prepared by: Adeyinka Fatona**

**1. Introduction**

The City of New York has initiated a Payroll Data Integration project to *centralize, transform, and analyse payroll data* across municipal agencies. The goal is to build a highly scalable, automated, and efficient data pipeline that enhances *financial oversight* and *public transparency* regarding payroll expenditures.

As a Data Engineer, my role was to design and implement a robust ETL (Extract, Transform, Load) pipeline leveraging Azure cloud technologies. This report outlines the technical architecture, data pipeline design, schema implementation, and key takeaways from the project.

**2. Project Objectives**

The NYC Payroll Data Integration project serves two primary objectives:

* **Financial Resource Allocation Analysis** – Assessing how financial resources are distributed, particularly for *overtime pay and budget allocations.*
* **Public Transparency and Accessibility** – Providing structured payroll data that can be queried to the public, ensuring transparency in taxpayer-funded salaries and overtime compensation.

To achieve these goals, I engineered a cloud-based data pipeline that ingests, processes, and stores payroll data efficiently.

**3. Tech Stack Selection**

To build a dynamic, automated, and scalable data integration pipeline, I chose the following modern cloud technologies:

*Core Technologies Used:*

* **Python** – For scripting data extraction, transformation, and validation.
* **SQL** – For efficient data querying, aggregation, and warehousing within Azure SQL databases.
* **Azure Data Factory (ADF)** – For ETL orchestration, automating data ingestion, transformation, and movement.
* **Azure Data Lake Storage** – For cost-efficient, scalable storage of raw and processed payroll data.
* **Pandas** – For advanced data manipulation and cleansing before ingestion into the warehouse.

These tools enable the pipeline to be fault-tolerant, optimized for performance, and scalable to accommodate growing payroll datasets.

**4. Data Pipeline Architecture**

The NYC Payroll Integration Pipeline follows a structured, end-to-end data engineering workflow, as illustrated below:

**Pipeline Flow:**

1. **Data Ingestion (CSV Source to Azure Data Lake)**
   * Raw payroll data is collected in *CSV format* from multiple municipal agencies.
   * Python scripts validate and clean data before ingestion.
   * Cleaned data is stored in *Azure Data Lake Storage* for scalability and cost efficiency.
2. **Data Transformation (Azure SQL Staging & Processing)**
   * Azure Data Factory (ADF) automates data movement from *Data*Lake to SQL Staging Layer.
   * Data is structured, deduplicated, and enriched in staging tables.
3. **Data Warehousing (Azure SQL Serving Layer)**
   * Validated data is moved to the *final serving layer*, structured for optimized querying and analytics.
   * This layer powers real-time analysis, BI dashboards, and financial reports.
4. **Analytics & Reporting**
   * The processed data is integrated into Power BI for payroll insights.
   * Reports provide a granular breakdown of salaries, overtime pay, and agency-specific budgets.

This pipeline ensures data integrity, high performance, and minimal latency in payroll reporting.

**5. Data Schema & Relationships**

The payroll data warehouse is designed using a star schema model to optimize data retrieval and analytics.

**Schema Overview**

* **Fact Table (payroll\_fact)** – Stores core payroll records, including base salary, overtime hours, and total payments.
* **Dimension Tables:**
  + **employee\_dim** – Contains employee details and associated agencies.
  + **agency\_dim** – Stores agency metadata, including start dates and boroughlocations.
  + **title\_dim** – Defines job roles and classifications.

This schema enhances query performance, ensures referential integrity, and enables efficient payroll analysis.

**6. Key Achievements & Insights**

🔹 **End-to-End Automation** – The pipeline eliminates manual processes, ensuring automated data ingestion, transformation, and storage.

🔹 **Scalability & Performance Optimization** – Using Azure SQL & Data Lake, the pipeline can handle increasing payroll data volumes without performance degradation.

🔹 **Enhanced Data Transparency** – The structured pipeline enables quick access to payroll records, improving budget accountability.

🔹 **Fault-Tolerant & Reliable Architecture** – The modular ETL design ensures that failures in any stage do not disrupt the entire workflow.

**7. Challenges & Solutions**

| **Challenge** | **Solution Implemented** |
| --- | --- |
| Potential Large Data handling and data change | Optimized SQL indexing for efficient querying. |
| Ensuring Data Accuracy & Integrity | Used Python validation scripts & foreign key constraints to maintain consistency. |
| Automating Pipeline Execution | Leveraged Azure Data Factory scheduling & monitoring for seamless execution. |
| Cost Optimization | Used Azure Data Lake hierarchical storage to reduce storage costs. |

These strategic solutions ensured a resilient, high-performance pipeline.

**8. Conclusion & Future Improvements**

The **NYC Payroll Data Integration** project is successfully build to be able to, ensure efficient payroll data processing, analytics, and transparency.

**Future Enhancements:**

* Real-Time Data Processing – Implementing streaming pipelines for real-time payroll updates.
* Machine Learning Integration – Applying predictive analytics to detect anomalies in overtime payments.
* Enhanced Data Access – Developing APIs for external access to payroll datasets.

This project lays a solid foundation for data-driven payroll management, setting the stage for advanced financial analytics and AI-driven insights.

This work highlights the power of data engineering in transforming raw payroll records into meaningful financial intelligence, ensuring accountability, efficiency, and data-driven governance.