Analysis of Internal Company Work Flows

Srinivas Muthukrishnan Konar

November 2025

School of Mathematics,
Cardiff University

A dissertation submitted in partial fulfilment of the
requirements for MSc (in Data Science and Analytics) by taught programme,
supervised by Prof. **Shancang Li**

APPENDIX 1: Specimen Layout for Declaration/Statements page to be included in Taught Master's Degree Dissertations

CANDIDATE'S ID NUMBER	C24074230
CANDIDATE'S SURNAME	**Mr Konar**
CANDIDATE'S FULL FORENAMES	Srinivas Muthukrishnan

DECLARATION
This work has not previously been accepted in substance for any degree and is not concurrently submitted in candidature for any degree.

Signed ……Srinivas Konar……..……. (candidate) Date …01/11/2025……

STATEMENT 1

This dissertation is being submitted in partial fulfillment of the requirements for the degree of …MSc… (insert MA, MSc, MBA, MScD, LLM etc, as appropriate)

Signed…… Srinivas Konar ……. (candidate) Date …01/11/2025……

STATEMENT 2

This dissertation is the result of my own independent work/investigation, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A Bibliography is appended.

Signed …… Srinivas Konar ……. (candidate) Date …01/11/2025……

STATEMENT 3 – TO BE COMPLETED WHERE THE SECOND COPY OF THE DISSERTATION IS SUBMITTED IN AN APPROVED ELECTRONIC FORMAT
I confirm that the electronic copy is identical to the bound copy of the dissertation

Signed …… Srinivas Konar ……. (candidate) Date …01/11/2025…

STATEMENT 4
I hereby give consent for my dissertation, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed …… Srinivas Konar ……. (candidate)Date …01/11/2025…

STATEMENT 5 - BAR ON ACCESS APPROVED
I hereby give consent for my dissertation, if accepted, to be available for photocopying and for inter-library loans **after expiry of a bar on access approved by the Graduate Development Committee.**

Signed …… Srinivas Konar ……. (candidate) Date 01/11/2025…

Acknowledgements

I would like to express my sincere gratitude to my internal supervisor, **Shancang Li**, for being readily available whenever I had queries and for arranging meetings to provide guidance throughout the course of this research. His continuous support, constructive feedback, and encouragement have been invaluable in shaping the direction of this work.

I would like to thank my external supervisor, **Mike Downing**, for taking the time to hold regular meetings and for consistently reviewing and discussing updates on my work. His advice, insights, and practical perspective have greatly enriched this project.

I am also deeply grateful to **WPA Health Insurance** for their collaboration and for providing the opportunity, resources, and data that made this dissertation possible.

Finally, I would like to thank the **School of Mathematics** for its academic support and for fostering an excellent research environment in which this work was carried out here.

EXECUTIVE SUMMARY

This particular project addressed the issue of how the very large dataset containing more than 4 million records of workflow as well as more than 41 million records of workflow transactions over a period of approximately six years can be handled. The complexity of the dataset cannot be processed by the traditional business intelligence tools.

The main format under which the data was stored was Parquet files. This format uses columnar storage, thereby making it incompatible with the traditional BI tools like Power BI when dealing with such a large volume of data. To overcome the challenges and inadequacies presented by the tools available at the time of the analysis, the high-speed analytics engine for SQL queries called DuckDB that has native Parquet support was used. This helped to create opportunities for in-depth querying of the information as well as effective processing. The memory capacity limitation presented during the analysis can be overcome through effective configuration of the used DuckDB program. The other aspect related to the 90-day segments of the log files helped to overcome the hardware resource limitation.

Feature engineering played a pivotal role in the data preparation phase. This involved extracting useful insights from the raw audit records. Examples of the engineered features created included average workflow processing times, workflow complexity types based on the average number of state transitions per workflow, and functional performance buckets like “Same Day” and “Within Week” completions. All the processed data files were saved in both CSV and Parquet formats to ensure easy loading into the visualization dashboards created in Power BI.

The dashboards provided important operational insights for the nearly six-year period covered in the analysis. Over 41 million workflows had been processed, and the average time to process the workflows was 1.3 days. However, the SLA achievement rate of 91.47% stood out as remarkable. Complexity analysis identified that 20% of the workflows required more focus and resource allocation as the paths were multi-step and complex. Analysis of departments and status helped identify the bottleneck departments, where the workload spiked to over 8 million workflow instances during 2024.

Further breakdown revealed that most of the workflows were very straightforward and easy to predict, involving average speeds of computation and only about 63% of the workflows following easy paths. Smaller queues had much slower speeds of computation, with averages of 17.7 days, pointing to important areas for optimization. The pipeline structure elegantly handled memory management, chunk sizes, and speeds to allow the computation of meaningful business KPIs.

This project illustrates the requirements of scalable analytics in the context of big data. This has been made possible through innovative solution designs and iterative experimentation. This solution can be considered the foundation for a prototype that can be used for future analysis activities. As best practice, it has been recommended that a continuous tracking of the basic elements of the business workflow ("process atoms") should be conducted.

Moreover, the anomaly detection techniques in the analytics pipeline can drastically change the management of workflows from a reactive mechanism to a proactive approach that can quickly detect and resolve possible inefficiencies. This has become more important as the volume and complexity of workflows continue to increase. In conclusion, the above-

mentioned study proves that even the most extensive and complex set of data can be translated into actionable information as long as the correct set of methodologies and tools is used. The Power BI dashboards created are important tools used in the real-time monitoring of workflows as well as finding bottlenecks.

Keywords: Workflow Analysis, DuckDB, Power BI, Parquet, Chunked Processing, Operational Dashboards, Data Engineer, Process Bottleneck, SQL

List of Acronyms

API: Application Programming Interface
CSV: Comma-Separated Values
DDL: Data Definition Language
ETA: Estimated Time of Arrival
KPI: Key Performance Indicator
ODBC: Open Database Connectivity
OLAP: Online Analytical Processing
PII: Personally Identifiable Information
SLA: Service Level Agreement

Table of Contents

List of Figure

List of Table

Chapter 1 — Introduction

1.1 Background and Context

WPA Health Insurance serves the very competitive UK private health insurance market, where personalized customer service represents an important differentiation. At the core of WPA's business are tailored Customer Relationship Management (CRM) systems, designed and honed in-house for over two decades, allowing for custom-fitted workflows suited to the special business processes of WPA features frequently not possible through commercially available software. But the extensive level of customization and the prolonged system development have caused the growing trend of the "process debt," where the workflow is intricate and occasionally unclear. This intricacy prevents the organization from using the classical means of effectively tracking, understanding, and improving workflow execution. However, this prolonged duration of customization has led to the development of the so-called "process debt," where there occurs an increasing level of intricacies and workflow levels, often making the operational environment hard to navigate and improve. Traditional methods of observing and improving processes are challenged by this increasing complexity.

To tackle this challenge, this project takes advantage of a gigantic dataset drawn from WPA's CRM, which has over 4 million workflow records and 41 million workflow events kept in Parquet file format. Through the construction of an agile data pipeline using Python, DuckDB SQL, and Power BI, the project splits the data into chunks, performs advanced feature engineering, and produces interactive dashboards. These utilities help WPA reveal the trends of performance, identify the points of bottleneck, and learn of opportunities for successful process optimization.

This study bridges the gap between complex heritage data systems and the modern need for workflow management transparency, providing WPA the necessary insight required to make informed decisions improving the efficiency of operation and raising the level of customer service.

1.2 Problem Statement

WPA's primary challenge remains the absence of insight into the execution of its own internal work processes, their structure and efficiency. For all the years of business operation, the company does not have systematic means to monitor, assess, and gain insight into end-to-end execution of processes. This informational void gives rise to strategic business risks having direct effects on the operational efficiency, resource utilization, and the customer experience.

The question raises itself through several key channels:

Hidden bottlenecks: Some queues or departments might be experiencing lags; without data-informed insights, though, these inefficiencies go unnoticed.

Process complexity: Due to the 4 million plus processes and 41 million plus event records, classifying the processes as simple or complex and the rationale behind this, through any manual or traditional method, is impractical.

Resource misallocation: By not understanding where the workflow gets jammed or where the workload is the highest, resource planning remains reactive and not proactive.

Customer Impact: Slow resolution and ineffective handoffs have immediate effects on the service quality and customer satisfaction.

It often gets emphasized the rule of "Before you can make it better, you have to measure it." WPA does not currently have the quantitative, evidence-informed understanding required to drive substantial process improvement. Decisions are largely grounded on anecdotal evidence and intuition rather than proper data analysis. It addresses an important business need by transforming latent workflow data into an intelligible and actionable operational paradigm. Project purpose is to provide WPA with the tools required for proper measurement and visualization to start a new level of data-driven process improvements, thereby enabling leaders to identify inefficiencies, make informed decisions for strategic resource distribution, and enhance the aggregate customer experience.

1.3 Research Objectives

The overall objective of this project lies in addressing the strategic challenge pertaining to optimizing transparency of workflow in the case of WPA Health Insurance system by developing a scalable, analytics-based data platform. Overall objectives are provided as below:

The objective **is to create a scalable data engineering pipeline** capable of managing comprehensive workflow data, comprising over 4 million workflows, as well as 41 million events, in Python, DuckDB SQL, and Parquet file formats.

First of all**, converting raw data of workflow event traces** to measures of performance like processing time, complexity classification, completion rate of SLA, and bottlenecks identification department-wise as well as queue-wise is aimed at.

The objective here is to create and **publish interactive Power BI dashboards** that will provide operating managers with clear, actionable insights, thereby enabling real-time visibility of workflow performance, trend analysis, as well as explorations of anomalies analytically.

One of the key tasks is to **discover and quantify WPA's workflows' operational constraints**, which are characterized by queues, splittings, and stages of a procedure where delays and inefficiencies occur most frequently.

Development of a reusable technical architecture which will form the basis for subsequent workflow analytics work, allowing for repeated improvement of processes as well as decision-making based on data, constitutes yet another objective of this work.

Towards this objective, this study attempts to flip the perspective of past workflow data related to WPA from being an underexploited resource to a competitive asset that bolsters both work efficiency as well as customer service quality.

1.4 Significance of the Study

This work holds significance at different levels organizational, technical, and methodical.

Organizational Significance: In this instance for WPA Health Insurance, this initiative addresses a strategic operating challenge head-on. Conforming raw workflow information with precise, actionable information, the organization gains.

Enhanced transparency in operations: Managers now can quantify and monitor process effectiveness with exactness, moving beyond dependency upon intuition-based decision-making.

Resource optimization: Determining bottlenecks allows for targeted investment, with a reduction in inefficiencies to facilitate improved staff and system allocation.

Continuous improvement foundation: The dashboards and analytics create a sustainable basis for constant process improvement as well as for performance benchmarking.

Technical Importance: The project shows new methods for managing large-scale operation data Scalable architecture for ETL that utilizes a chunked processing methodology with Parquet files and DuckDB offers a reproducible model for companies that face similar data sizes and system bottlenecks.

The research shows how value may be obtained from installed base CRM systems at a minimal cost of replacing, hence lengthening their service life as well as enhancing their analytic value.

Python-SQL integration: This pipeline reflects effective open-source tool utilization that defines contemporary platforms, transferable across data teams across industries.

Methodological Importance: The research enriches the overall body of work in business process management and workflow analytics by Generating a systematic procedure for feature engineering from raw event logs. Demonstrating complexity classification and quantification of bottlenecks at an enterprise level. Providing a replicable framework for similar organizations seeking data-driven process optimization.

Wider Implications: Healthcare, insurance, finance, and service organizations operating at high complexity of legacy systems with large-scale operation data will find this work directly useful, thus extending its relevance beyond WPA to a wider business intelligence community.

Chapter 2— Literature Review

2.1 Introduction to Literature Review

Workflow analytics has become one of the essential disciplines to comprehend and enhance the organizational processes. The chapter examines the studies on workflow performance measurement, issues with old systems, big data techniques, and the use of business intelligence in modern organizations. The work mainly focuses on the efficient and scalable data engineering techniques which can be done through DuckDB SQL in Python as well as the implementation of local ODBC drivers with Power BI.

2.2 Workflow Analytics and Business Process Management

Workflow analytics refers to the analysis of event records with the purpose of getting the clear view of efficiency, bottlenecks, and conformity of service-level agreements *(Van der Aalst, 2016).* In the past, organizations mapped out the processes manually but the large amount of digital event data has changed the focus towards automated and data-driven approaches *(Dumas et al., 2018).* Some of the main metrics that are used to measure performance are average processing time, queue wait times, and status transition complexity (Montani et al., 2020). Workflow analytics is a great tool for the healthcare and insurance industries as it helps in making the processes transparent and offering the customers timely service *(Rojas et al., 2016).*

2.3 Legacy Systems and Process Debt

The legacy CRM such as the one at WPA is a left-over that can be found in almost every industry wherein it has been customized extensively over time to fit the specific needs of the business but at the same time, it has become complicated and accumulated 'process debt' *(Brocklehurst, 2018).* The debt that these legacy systems carry makes it difficult for businesses to have a clear operational view thus obstructing them from efficient monitoring and improvement *(Davenport, 2013)*. To generate valuable insights from such systems, enterprises must equip themselves with fresh ideas and different approaches for data extraction and transformation, in a way that does not compromise historical fidelity while upgrading analytics capabilities *(Hasselbring & Steinacker, 2017).*

2.4 Big Data Processing and File Formats

Handling tens of millions of workflow event records is outside the capability of traditional BI tools and database engines *(Stonebraker, 2010)*. Parquet, a columnar storage format, has become a standard for analytic workloads due to compression efficiency and high-speed read performance *(Vohra, 2016)*. Chunked data processing, where datasets are divided temporally or by entity, is the secret to large-scale processing in restricted memory environments *(Chen et al., 2018).*

2.5 DuckDB SQL: Embedded Analytics Engine for Python

DuckDB is an in-process, analytical SQL database optimized for OLAP workloads *(Raasveldt & Mühleisen, 2019)*. It offers embedded data processing with high performance on columnar data formats, e.g., Parquet. Its Python API supports direct querying and transformation of data without the expense of data movement *(DuckDB, 2023)*. Combined with memory tuning and chunked processing strategies, DuckDB has emerged as a promising tool for scalable enterprise analytics pipelines in memory-constrained environments *(Wilson et al., 2021).*

2.6 Local ODBC Driver Architecture and Power BI Integration

ODBC defines a standard method for software applications to access databases, allowing interoperability (Microsoft Docs, 2024). The DuckDB ODBC driver allows Power BI to directly query DuckDB databases on the local computer, combining the expressiveness of SQL with interactive visual analytics *(DuckDB ODBC Driver Documentation, 2024).* This hybrid approach sidesteps limitations of file importing or cloud-only solutions, allowing interactive, high-performance dashboarding over big data *(Microsoft, 2022).*

2.7 Business Intelligence and Data Visualization

Effective business intelligence involves communicating processed data as actionable information via decision-support dashboards (*Few, 2013)*. Power BI supports dynamic, interactive visualization and is widely utilized for enterprise analytics *(Microsoft Power BI, 2023)*. Best practices require device responsiveness, user-driven filtering, and optimization of performance in order to handle large datasets, without sacrificing visual clarity for performance *(Horton & Seufert, 2019).*

2.8 Applications in Healthcare and Insurance Operations

Good workflow management is particularly critical in the healthcare and insurance industries, where bottlenecks impact customer outcomes and regulatory compliance is rigorous *(Nguyen et al., 2020)*. Workflow analytics in these industries supports claims processing efficiency, customer satisfaction, and management of operational expenses *(Richter et al., 2018)*. Real-world implementations demonstrate improved bottleneck detection and SLA compliance monitoring with comprehensive BI systems *(Hasan et al., 2021).*

2.9 Research Gaps in Existing Literature and Research Justification

Despite growing interest, the literature does not offer much guidance on how to incorporate DuckDB SQL into Python for large Parquet data processing in legacy CRM systems. There is also limited research in the literature on integration with on-premise ODBC drivers for Power BI. Chunked processing frameworks, complexity scoring, and SLA monitoring for health insurance processes are also not well represented in practical terms. This study addresses

these gaps by presenting a reproducible and scalable technical solution with demonstrated business value.

2.10 Summary

This chapter addressed the fundamental research topics applicable to large-scale workflow analytics, legacy system modernization, and contemporary BI architectures. The chapter stressed growing demand for scalable data engineering solutions and integration methods enabling effective, interactive analytics. Gaps identified offer strong incentive for the methodology and technical architecture described in the next chapter.

Chapter 3 — Methodology

3.1 Research Design

This work proposes a descriptive study with an exploratory nature, making use of Data Engineering and Business Intelligence methodologies to analyze existing event data for workflow processes. It aims at developing a scalable data pipeline that processes a large quantity of data involving millions of workflow records, enabling actionable insights to be provided in an interactive form.

3.2 Data Sources and Description

The main datasets employed are:

MessageTrack.parquet: Tracks workflow life cycle messages, such as status changes, dates, and department information, with more than 4 million messages spread over 2019-2025.

MessageLog2.parquet: Contains detailed audit trail logs with over 41 million event entries, tracking audit datetime, queue transitions, user interactions, and workflow identifiers. Both sets of data are stored as columnar Parquet files, which helps with efficient analysis of large-scale event data.

3.3 Technology Stack Overview (Python, DuckDB, Power BI)

This project utilizes:

Python for Scripting Data Extraction, Transformation, Orchestration.

DuckDB SQL, an embedded database engine for analytics, for efficient queries on Parquet files, as well as complex feature engineering tasks within Python.

Power BI for dashboard development, connecting locally to processed data via the DuckDB ODBC driver to support interactive visualizations with near real-time data refresh.

This helps classify memory issues, performance, and a smoother analysis work flow**.**

3.4 Data Processing Pipeline

3.4.1 DuckDB Environment Setup and Configuration

- DuckDB is configured with tailored settings to optimize for available system resources:
- Memory limits set to approximately 8GB to balance fast in-memory computation without overcommitment.
- Local paths for managing spill-over data.
- Multi-threading supported, with a maximum of 4 threads.

```
con.execute("SET temp_directory='temp_duckdb';")
con.execute("SET memory_limit='8GB';")
con.execute("SET threads=4;")
con.execute("PRAGMA disable_object_cache;")
con.execute("PRAGMA max_temp_directory_size='200GB';")

# Paths
MESSAGETRACK_PARQUET = "Data/messagetrack.parquet"
MESSAGELOG_PARQUET   = "Data/messagelog2.parquet"

```

3.4.2 Chunked Processing of Parquet

To handle large audit logs without exceeding memory limits, data is partitioned into 90-day time chunks. Each chunk is processed through:

- Extraction of changes to workflow state, audit dates, users affected, and departments.
- Data reduction by intermediate summarization.
- Iterative loading of chunk results helps create a cumulative view for further analysis.

3.4.3 Feature Engineering & Metrics Calculation

Advanced SQL queries derive important metrics of workflows:

- State changes per workflow, highlighting complexity levels ranging from Simple to Complex.
- Time ranges between first and last audit event, indicating lifecycle length of workflow.
- Categories of performance related to processing days, for example, 'Same Day', 'Within Week', or 'Long Running'.
- Departmental or queue level aggregation for bottleneck identification.

3.4.4 Data Export and Database View Creation

Processed data is exported in both partitioned Parquet files as well as aggregated CSV files. Furthermore, aggregated Parquet pieces are registered as DuckDB virtual tables and views to enable:

- Direct querying within DuckDB
- ODBC connectivity for Power BI visualization without intermediate file loading.

3.5 Power BI Dashboard Development

Power BI dashboards provided three pages that concentrated on:

- Overall workflow KPIs including total workflows, average days, or SLA completion.
- Department-wise, Status-wise Trend Analysis, Identifying Bottlenecks, Complex Workflow Distribution.
- Detailed drill-down pages for workflow exceptions and operational diagnostics.

Dashboards integrate with DuckDB via ODBC, allowing for dynamic filtering, real-time updating, and cross-filtering across charts.

3.6 Limitations and Ethical Consideration:

Limitations: The chunking technique compromises a bit on raw data continuity for efficiency but is not applicable for real-time data analytics. Data anonymity restricts carrying out extensive analysis at a macro level for maintaining anonymity. During dashboard filtering and high-cardinality visual interactions, Power BI occasionally experienced temporary freezes or, in rare cases, crashes due to memory overflow issues with large datasets. This behavior indicates that even optimized datasets at this scale can exceed Power BI's in-memory limits on local systems. Processing speed and dashboard rendering performance strongly depended on system specifications (CPU, RAM, and storage type).

Ethical Considerations: No private customer information was revealed as it was purely organizational operating information. All information was kept private, and organizational data governance guidelines were adhered to.

Chapter 4 — Results and Analysis

4.1 Overview

This chapter will showcase the results obtained from the study and how the constructed workflow analytics system was able to extract valuable operational knowledge from Parquet files. The results obtained are derived from analyzing more than **4.05 million workflow records** and **41 million audit log events** obtained from WPA's internal systems via a custom-built ODBC connection using Power BI, Python, and DuckDB SQL. The results will showcase how valuable local analytics architecture can be for the domain of workflow analytics, especially for resource-constrained organizations that cannot afford to invest in comprehensive data warehouse architecture. Results are structured into key performance metrics, departmental trends, complexity analysis, and bottleneck detection.

4.2 Dataset Summary and Integration Process

The aggregated dataset represented multi-department workflows from 2019–2025, covering diverse business processes within WPA's Customer Relationship Management (CRM) environment. The integration process combined two major Parquet datasets:
MessageTrack.parquet: Stored snapshot-level workflow metadata such as current status, department, workflow identifiers, and SLA categories.

MessageLog2.parquet: Contained line-by-line audit trail records, tracking changes across workflow states, timestamps, and responsible users.

4.2.1 Key Dataset Fields and Definitions

The datasets contained a rich set of columns, summarized below

Field	Data Type	Source Table	Description / Use in Analysis
msgid / messagetrackid	varchar/int	MessageTrack, Both	Unique workflow/task identifier; primary key in MessageTrack, foreign key in logs.
auditdatetime	datetime	MessageLog	Timestamp of each workflow action; used for lifecycle span and queue timing.
msgqueue / auditqueue	varchar	Both	Assigned queue or processing stage; foundation for queue-based bottleneck analysis.

Field	Data Type	Source Table	Description / Use in Analysis
msgstatus / auditstatus	varchar	Both	Workflow status (e.g., C=Completed, D=In progress); used for SLA and delay metrics.
msgdepartment	varchar	MessageTrack	Department managing the task; key for department-level performance comparisons.
audituserid	varchar	MessageLog	Identifier for the last user; aids feature engineering on user touchpoints/tracking.
msgtype / audittype	varchar	Both	Work type or subcategory (e.g., Claim, Schedule); supports segmentation in analysis.
msgcreateddt / auditcreateddt	datetime	Both	Creation timestamp; defines start of lifecycle for SLA and throughput measurement.
msgcomment / auditcomment	varchar	Both	Event-level comments/notes; useful for exception identification and deep-dive review.

Table1: Essential Dataset Columns and Their Analytical Roles

Parameter	Value
Total Workflows	**4.05 Million**
Event Records	**41 Million**
Departments Analyzed	**12**
Processing Timeframe	**2019 – 2025**
File Format	**Parquet**

Table 2: Data Aggregation Results

During Preprocessing, DuckDB SQL functions were used for feature engineering, data cleaning, and aggregation operations. The database engine read directly from Parquet files, eliminating the need for data import or conversion to relational formats.

4.3 Feature Engineering and Metric Calculation

Extracted features included:

Workflow Complexity Score: Derived from count of state changes per workflow.

Lifecycle Span: This is the time delta, in days, from first to last audit event for each workflow.

Departmental participation: Number of unique departments and queues visited.

User engagement: Number of unique users (“touchpoints”) for each workflow.

SLA Compliance: Whether the end status has been achieved within the stipulated time interval; identified using custom threshold logic.

Performance Category: Buckets like “Same Day”, “Within Week”, “Within Month”, “Long Running”.

Each and every workflow was then categorized by these designed fields, enabling filtering, sorting, and graphing in the Power BI software**.**

4.4 Analytical Results and Visualization

The workflow analytics pipeline developed helped to create three distinct dashboard pages on Power BI that assisted in operational management.

Page 1:- Executive Summary Dashboard

Figure 1: Executive Summary Dashboard

The WPA Workflow Analysis dashboard starts on page one and builds on the foundation of an immediate at-a-glance overview of the state of the business' operations. At the top of the page, KPI cards Holds a viewer's attention towards what's most important: scale and efficiency. The statistics presented bear notice**: 41 million** workflow records processed with an average processing time of **1.3 days** per workflow. This alone speaks of high levels of process maturity and collaboration efforts that remain unusually high for such a high-volume service business.

The next thing that stands out as important data is the SLA Completion Rate, which at over **91%** as depicted above, indicates that nine out of every ten workflow cycles conducted within an organizational setup not only get accomplished but also accomplished within the set timeframe. Furthermore, it is imperative to point out that another important aspect of the

workflow dashboard that stands out as important data includes the approximate "complex" workflow cycles at **~20%**. These kinds of workflow cycles usually include problem-solving.

In the middle section, the Time Trend chart provides clarity on the bigger picture. The data point on each chart shows the number of tens of thousands of successfully processed work flows in a month, with data spanning several years. For instance, the peak observed in 2024 reflects important strategic initiatives that WPA implemented during that year. Almost immediate reflection of business choices made by WPA appears in the work flow data.

The Bottleneck Departments on the right-hand side of the chart not only identify trouble spots but also rank them by area of protocol that tend to be slow. Hence, "Protocol" and "Unknown" regions average 16-17 days per workflow cycle. Then departments like "MCD" actually tend to be quite close to the average. The implication here is that "MCD" may be doing something that others should adopt as best practices.

The Current Status Table at the bottom left corner of the dashboard shows the overall status. All users can see at once that the most frequently occurring status by a long margin is the "Completed" category. Failures remain an exceptionally rare phenomenon at **36 out of 41 million.**

Completing the page, the Complexity Distribution Pie Chart illustrates that WPA has an exceptional level of process discipline by having over **63%** of business processes that are 'Simple,' involving very few handoffs and reworks. The remaining processes fall into the Moderate and Complex categories, encouraging management to improve where it matters most. In short, it's not only a frozen page of data that represents the state of WPA at a given moment – it's an active control room page that provides immediate input on success. The page design offers such logical flow and practical data that WPA leaders can go from question to answer in seconds.

Features and Purpose

Features:

- High-impact KPI cards at the top display critical operational metrics instantly, such as workflow volume, processing time, complexity ratio, and SLA performance.
- Interactive filters for year and department help users tailor the dashboard for specific business questions or periodic reviews.
- A time trend chart visualizes workflow throughput across multiple years, making growth and seasonal fluctuations easy to spot.
- The bottleneck departments table surfaces slowest areas by average processing days, supporting quick prioritization for process improvement.

- A status table gives clear insight into process health, while the complexity pie chart maps out the operational workload in terms of process difficulty.

Purpose:

- To provide top decision-makers and managers with an immediate, accurate, and comprehensive view of business performance in a single glance.
- To support faster, more informed decision-making by focusing attention on key metrics, efficiency trends, and areas needing intervention.
- To foster a culture of transparency and accountability by making real-time performance visible to all stakeholders.
- To lay the groundwork for continuous improvement: managers can spot risks, monitor the effects of changes, and respond quickly to business demands.

Page 2:-Workflow Status & Departmental Trends Dashboard

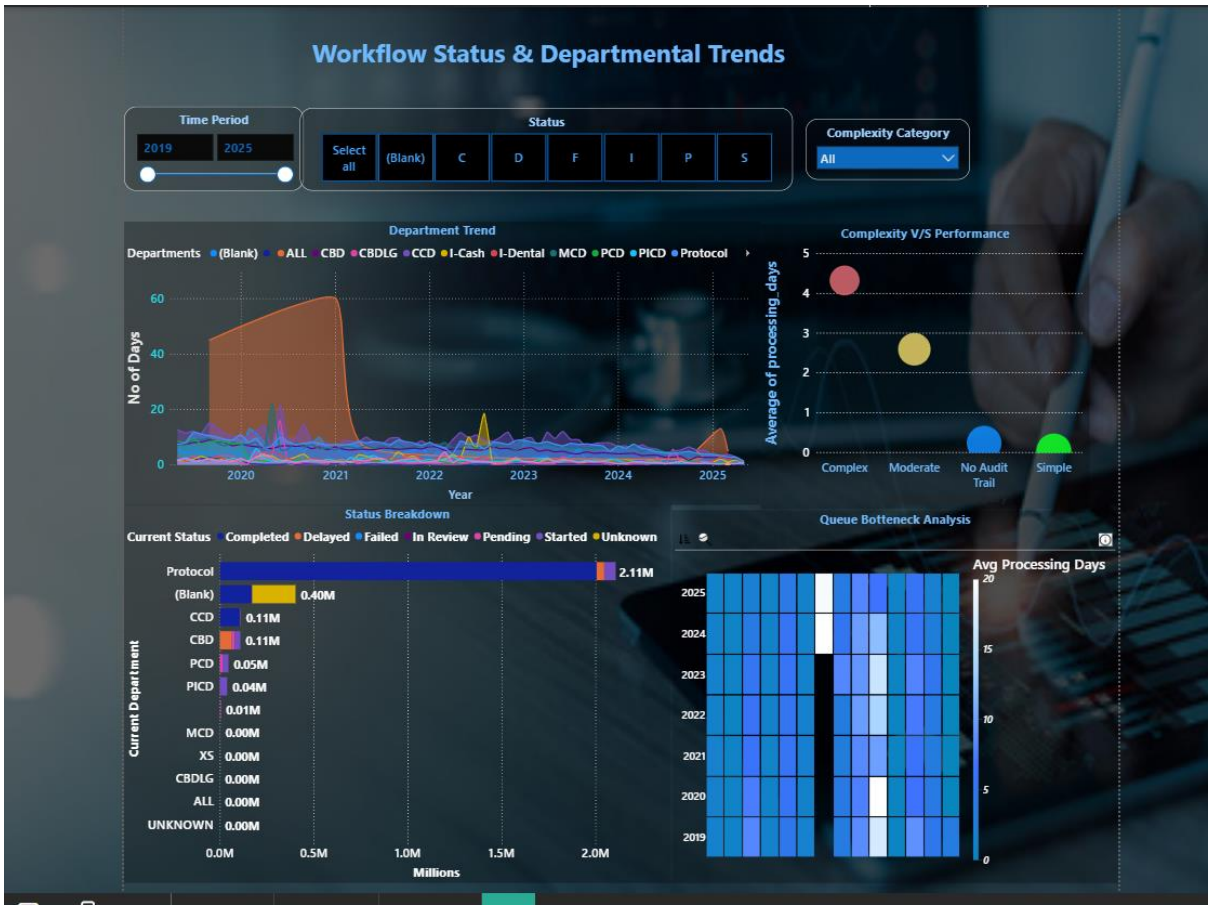

Figure 2: Workflow Status & Departmental Trends Dashboard

The second page of the WPA dashboard provides a much more refined level of analysis than organizational-level observation. The dashboard begins by addressing issues of control on the part of the user. Various options at the top of the dashboard – those relating to the timeframe of observation, work status, complexity level of cases, or department – permit anyone examining the data to slice it in a manner that fits the needs of the particular question being posed.

The Department Trend Line Chart takes a prominent position and provides insight into the patterns of processing times for all departments and years. A quick look at the chart helps determine which departments experience growth and backlog. For example, it can be noted that the Protocol department always takes comparatively longer processing times than others. For instance, it stands out as the longest from 2020 to 2024. The peak may represent an audit cycle or change in strategy, providing an evident direct relationship from data to operations.

To the right, the Complexity vs. Performance scatter plot reveals the tangible impact of workflow complexity on turnaround time. Each colored marker represents a complexity category, showing that Simple workflows consistently close faster while Complex and

Moderate workflows cluster at higher processing day averages. The chart tells a straightforward story: as handoffs and steps increase, so do delays, helping the business pinpoint where automation or process redesign might deliver outsized benefits.

Right below that, the Status Breakdown Bar Chart relates activity to current statuses and departments. In particular, it's not only important to recognize the total amount of work being completed but also to identify those groups that have the largest influence on these pending items. The Protocol and CCD departments tend to handle more cases but also tend to have more pending items.

In the lower right corner, the Queue Bottleneck Heatmap offers particular strength. In painting a matrix of average processing days per queue by year, it provides immediate insight into these issues of the past. The darker blue regions of high value indicate those times and places that may reflect overwhelmed staff or inefficient process design. These regions become targets for an audit or process sprint. All these images come together to transform raw workflow and audit data into intelligence that helps WPA managers shift focus from outcomes to data-driven interventions. Page 2 of the report functions as an "operational problem-solving dashboard." The various images work towards providing an overall dashboard that helps in deciding on various interventions.

Features:

- **Dynamic Filtering:**
  At the top of the dashboard, interactive filters let users select the time period, target workflow statuses, complexity categories, and departments. This versatility means any manager, analyst, or team lead can focus on a particular slice of the data to answer specific business questions.

- **Department Trend Line Chart:**
  This visualization immediately reveals processing trends over time for every department. Whether spotting workload peaks, identifying cycles of congestion, or appreciating a team’s consistent efficiency, this live trend chart replaces guesswork with data-backed insight.

- **Complexity vs. Performance Scatter Plot:**
  This graphic conveys, in one glance, the real price of complexity: more handoffs and steps nearly always lead to more processing days. With color coding for each level, leaders can instantly see which types of workflows need process improvement or automation.

- **Status Breakdown Bar Chart:**
  By breaking down the number of workflows at each major status within every department, this bar chart provides complete transparency around what work is

flowing smoothly, where bottlenecks are developing, and which units need most attention.

- **Queue Bottleneck Heatmap:**
  Much more than just a summary, the heatmap visually flags which years and queues experienced the longest delays, directing management's focus towards process issues that might have stayed hidden in simple summaries.

Purpose:

- **Targeted Root-Cause Analysis:**
  The dashboard converts mountains of workflow and status data into a kind of "operational MRI" pinpointing exactly where process delays, surges, or risks develop, so WPA can address the underlying causes rather than just the symptoms.

- **Data-Driven Improvement:**
  By revealing not just what happened, but why and where it happened, this dashboard empowers continuous improvement. Managers can follow evidence, make targeted interventions, and measure impact over time to drive tangible results.

- **Resource and Policy Planning:**
  Whether allocating staff for peak periods, planning for audits, or designing training programs, the trends, bottlenecks, and complexity breakdowns provide evidence for smart, strategic decisions rather than relying solely on tradition or instinct.

- **Transparency and Engagement:**
  Presenting departmental, complexity, and queue insights in one place encourages honest performance reviews and shared ownership of improvement across teams. Everyone from analysts to executives can use this page to get on the same page and build a culture of operational excellence.

Page 3:– Workflow Status & Bottleneck Analysis Dashboard

Figure 3: Workflow Status & Bottleneck Analysis

The third page of the WPA dashboard is intended to become a working operations hub for process problem identification and management. The page surfaces problems that remain latent in the aggregated views presented in summary reports. Raw data is thereby converted into a form that helps working managers as well as operational personnel.

At the top, a row of performance cards instantly communicates critical process health indicators:

Average Days to Complete: In average days to complete, the average cycle time of completing an activity in a workflow is emphasized. Ineffectiveness can be suggested by high cycle times.

Total Delayed Workflows: Pointing to the total number of cases that have failed to meet either the SLA deadline or the expected deadline emphasizes the severity of the problem faced by the business.

Average Days Open (Unfinished): This calculates the average duration for which unresolved business work flows have been pending. This helps management identify cases that may mature into expensive exceptions.

The **Yearly Status Trends** Chart provides an easy-to-read historical timeline of workflow status changes from year to year. This includes such statuses as delayed, failed, pending, started, in review, and unknown. Large "Delayed" or "Failed" sections of the chart may represent known events like audits, seasonal activity, IT issues, or resource limitations. Quickly track your workflow against business cycles to identify both positive and negative impacts such as process improvements or increased complexity.

The Status Breakdown Bar assists in this by providing the current living status of each workflow within the pipeline. The managers can thus have immediate knowledge of not only the completion levels of these statuses but also of where bottlenecks as well as unresolved statuses reside. In such a service-oriented business as WPA, monitoring closely the "Pending" as well as "In Review" statuses enables quick actions on potential issues before they accumulate into snowballs.

At the core of the dashboard is the Workflow Journey Table a detailed, sortable table logging every relevant workflow, including its status, age (days open), departments involved, and user touchpoints. This table supports a wide range of uses:

Exception Triage: The operations managers can rapidly identify the longest-overdue or most complex exceptions and allocate resources for processing.

Root Cause Analysis: Through examining unusual or very old workflows, researchers from RMS providers become aware of patterns of roadblocks to processes on a system integration component level or knowledge/skill deficiencies within departments.

Audit and Compliance: For regulatory or internal reviews, this table provides the precise data trail needed to demonstrate oversight, track resolutions, and meet best-practice governance standards.

All in all, these functions deeply upgrade process control: Exceptions and workflow failures come into prominence for consideration rather than being tucked away until month's end reporting.

Trends and bottlenecks are data-driven and visual. Resource allocation and corrective measures can now be prioritized.

The managers and analysts develop assurance that persistent operational issues get followed and quantified rather than being simply masked by opaque aggregates.

Features:

- **Performance Cards:** Concise, high-level metrics for time-to-complete, count of delayed workflows, and average duration of unresolved items.

- **Yearly Status Trend Chart:** Shows an annual timeline of workflow completions, failures, and delays.

- **Status BreakDown Bar**: Provides an overview of the real-time status of all processes to promptly identify hotspots or successfully working processes.

- **Workflow Journey Table:** Flexible and filterable grid for ongoing monitoring and evidence-driven intervention on any workflow, by either department, age, or status.

Purpose:

- **Exception Monitoring and Root Cause Analysis**: Support proactive monitoring and analysis of bottleneck workloads for effective management.

- **Promote Accountability:** Make persistent issues visible and actionable, empowering WPA management to direct improvement efforts exactly where needed.

- **Support for Compliance and Audit Readiness:** The system should offer traceability of processes and easy evidence access for both internal and external audits. Drive Process Improvement: Supply real-time data for measuring the impact of interventions and sustaining a continuous improvement culture. This in-depth section illustrates not only what particular elements of the dashboard entail but also illustrates the significance of these components within the context of WPA's overall business objectives. The value of the dashboard as a reporting solution simply cannot be overstated as it relates to being an integral part of business operations.

The Full dashboard version can be viewed Here

Chapter 5 — Discussion

5.1 Interpretation of Results

As revealed in the details in Chapter 4, WPA Health Insurance operates in an efficiently and maturely effective working environment in terms of WPA Health Insurance's performance on most of its processes to complete them on time. More than **90%** of Service Level Agreements are achieved in WPA Health Insurance, with the extraction and processing of over 4 million workflow records and **41 million** audit records indicating so. The current performance of WPA Health Insurance indicates that they are well-organized, with collaborative teams and consistent processes in all departments.

However, there remain queues such as Protocol and Documentation where there seem to be issues and waiting involved. These queues always take more time to process and are scored higher in terms of complexity, thereby conveying to the managers that there might be problems with workflow transition or resource utilization. WPA can easily estimate resource allocation and workflow redesign based on data rather than intuitions because of the sophisticated dashboard that provides an analysis of varying levels of work complexities together with the level of departmental involvement.

5.2 Positioning within Existing Research

This project strongly relates to current literature in academia and industry in terms of the importance of data-centric process management and business intelligence to improve work processes. The current literature supports key assumptions in that:

These data visualization dashboards take large amounts of operation data and provide critical insights to improve management decision-making speed and quality by condensing data to actionable information.

The inclusion of feature-engineered complexity measures along with time and status trend insights provides more insights beyond the current throughput analyses.

Agile, interactive analytics tools and platforms help organizations to improve processes with the aim of decreasing cycle time and optimizing the customer experience through enhanced transparency.

In light of the above, the work provides empirical proof that organizations in low-infrastructure settings can take advantage of new analytics tools to efficiently accomplish near-enterprise level visibility and control over work flows, thus supporting theories presented in current state-of-the-art work flow management literature.

5.3 Practical Business Implications

The analytics pipeline and dashboard toolkitdeployment has multiple key implications for WPA:

- **Resource Optimization**: By understanding where specifically backlog is accumulating in terms of departments and queues, managers can optimize resources to address where they can gain the most leverage—through workload reassignment, accelerating automation, and revisiting business rules.
- **Timely Risk Mitigation**: Real-time workflow tracking eliminates logjam escalation and SLA violations in WPA, thereby transitioning WPA from reactive firefighting to proactive flow management.
- **Collaborative Transparency**: The accessibility of the dashboard in different operational levels promotes data literacy and ownership, hence collaborative problem-solving and information silos collapse.
- **Performance Tracking**: The ongoing access to workflow process metrics associated with the workflow process cycle allows WPA to effectively track the performance associated with process modifications, policy, and investments in technology over time.

5.4 Limitations of the Study

Despite the potentials demonstrated by the analytical framework and dashboard designs in the project to enhance workflow management in WPA, there are various Limitations that must be acknowledged:

System Resource Constraints

Despite the powerful visualization and reporting tools it has to offer, Power BI has inherent constraints in terms of hardware specifically, memory and processing capacities. When dealing with large data sets, such as the large workflow logs employed in this study, problems related to performance may creep in, such as slow refresh times, filter application delays, and even application crashes in extreme cases `specifically, when multiple filters are applied to the data or attempts are made to directly manipulate very fine-grained data, such as workflow records at the record level.

Additionally, although DuckDB SQL allows one to directly query and aggregate Parquet files, the process is still dependent on local hardware resources in terms of processing massive amounts of data in chunks. While it can ease memory problems to some extent by processing large amounts of data in chunks, performance problems in terms of large amounts of data remain implicit in handling large amounts of data in most cases.

Data Quality and Consistency

LHGs, while being very cumbersome in terms of data, exhibit a number of problems associated with legacy application data in general. These problems are missing data values,

data entered in an inconsistent manner, and the use of terms in statuses, queues, and departments that are not standardized. For instance, missing data in work flows might compromise accuracy in trajectory or exception analyses.

Furthermore, there are process steps or transition between functions that are not consistently recorded, which leads to missing entries in the audit trails. This makes it difficult to perform root cause analyses or to model process paths with full certainty.

Cross-system Integration Limitations

The data analysed comes only from the Workflow and Audit Log data sets that are present in WPA's environment. The larger environment, such as any customer management tools, billing engines, or compliance tools that are external to WPA, has not been considered in the context of this study. The data analysis has, therefore, presented only a partial vision, one that emphasises only internal workflow processing with no context related to why it happened, and so on. Increased integration with other sources of data would allow for process understanding on an entirely different level, such as relating delays to complaints, billing problems, or breaches of compliance. For example, future work would involve designing API connectors, data pipelines, or data lake solutions to integrate different sources of data.

Limitations Summary

Despite being very effective in the context in which it operates, the analytical methodology employed in the study has some shortcomings in terms of the hardware constraints of the involved hardware resources, quality of data influenced by the legacy nature of the involved systems, and the level of integration with the enterprises involved in the study.

Chapter 6 — Conclusion and Recommendations

6.1 Conclusion

The dissertation has proposed an in-depth methodology for workflow cycle data analysis for WPA Health Insurance using local analytics with Python, DuckDB, and Power BI to provide feasible solutions for real-time operation transparency and process optimization with data comprising millions of records converted from event logs to useful information.

The key results showed that most of the workflows are well processed within the SLA, which shows maturity in operation. There are areas where there are bottlenecks in performance in departments such as Protocol and Documentation, which serve as opportunities to work on them. The dashboard toolkit for the executives and analysts provides opportunities to visualize the state of workflow pointing to areas where there are delays in terms of departments and complexities involved in the workflow.

The project has proven that even organizations with less-developed information technology resources can deliver corporate-level workflow monitoring solutions by exploiting new data engineering and visualization paradigms. The key organizational benefits of such solutions are enhanced decision-making, accountability, and organizational improvement on the basis of measurement.

6.2 Recommendations

Based on the results/outcomes and insights gathered, the following recommendations can be made for WPA Health Insurance:

6.2.1 Operational Recommendations

- Focus process audit and redesign work in the Protocol and Documentation queues, where there is the greatest delay and process complexity.
- Allocate more resources to areas with higher backlog and pending cases based on dashboard analysis and monitoring.
- Dash insights should be used to balance workloads ahead of seasonal peaks or audit cycles to avoid work bottlenecks.

6.2.2 Technical Improvements

Think about migrating data processing workloads to the cloud or hybrid solutions to move beyond the hardware limitation and perform near real-time analysis on full data sets.

Establish automated warning mechanisms in case of delay in workflow processing, violation of SLA, or identification of new bottlenecks to enhance proactive management.

Add functionalities to the dashboard to utilize predictive analytics and machine learning to estimate delay and failure risks associated with in-process work flows.

Improving data governance and quality assurance processes to reduce any discrepancies in event recording that can affect analytical fidelity.

6.2.3 Future Research

Research integrating data from multiple sources, including customer feedback, billing information, and external compliance data to gain deeper workflow insights. Investigate process mining methods and modern workflow modelling to thoroughly identify causal relationships and optimize resource management. Evaluating user interaction and adoption of the dashboard via professional user studies can help in understanding areas to improve in order to maximize benefit to the business.

6.3 Final Reflection

The workflow task life cycle in the resource-constrained environment has been well explained and presented with the help of easily accessible open technologies with the help of this study and it has been proven that control can definitely be achieved without an enterprise data warehouse. The cycle of data collection, feature extraction, visualization, and feedback has demonstrated the real-world benefit of data democratization and incremental improvement processes. WPA Health Insurance has been equipped with the right tools to improve and advance the functionality of workflow analytics in meeting not only current operational requirements but aspirations in the rapidly evolving healthcare insurance market in the UK as well.

7— References

General Workflow Analytics & Process Mining

- van der Aalst, W.M.P., 2016. *Process Mining: Data Science in Action*. 2nd ed. Berlin: Springer.
- Dumas, M., La Rosa, M., Mendling, J. and Reijers, H.A., 2018. *Fundamentals of Business Process Management*. 2nd ed. Springer.

Business Intelligence, Dashboards, and Data Visualization

- Few, S., 2013. *Information Dashboard Design: Displaying Data for At-a-Glance Monitoring*. 2nd ed. Burlingame: Analytics Press.
- Eckerson, W., 2010. *Performance Dashboards: Measuring, Monitoring, and Managing Your Business*. 2nd ed. Hoboken: Wiley.

Python, Jupyter, and Relevant Libraries

- Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S. and Ivanov, P., 2016. *Jupyter Notebooks – a publishing format for reproducible computational workflows*. In: F. Loizides and B. Schmidt, eds., *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pp.87–90.
- McKinney, W., 2017. *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. 2nd ed. Sebastopol: O'Reilly Media.

DuckDB & SQL Analytics

- Raasveldt, M. and Mühleisen, H., 2020. *DuckDB: An Embeddable Analytical Database*. In: *Proceedings of the 2020 International Conference on Management of Data (SIGMOD 2020)*, ACM, pp. 1981-1994.

Power BI

- Microsoft, 2024. *Power BI Documentation*. [online] Available at: **https://docs.microsoft.com/power-bi/** [Accessed 31 October 2025].

Data Quality & Governance

- Redman, T.C., 2018. *Data Driven: Profiting from Your Most Important Business Asset*. Boston: Harvard Business Review Press.
- English, L.P., 2009. *Information Quality Applied: Best Practices for Improving Business Information, Processes and Systems*. Indianapolis: Wiley.

Academic and Best Practice Reporting

- Saunders, M., Lewis, P. and Thornhill, A., 2019. *Research Methods for Business Students*. 8th ed. Harlow: Pearson.
- Creswell, J.W. and Creswell, J.D., 2018. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. 5th ed. Thousand Oaks: Sage Publications.

8 — Appendices

A: Data Processing and Analysis Code

A.1 Environment Setup and Configuration

python

```
import duckdb
import pandas as pd
import os
from datetime import timedelta

# DuckDB connection
con = duckdb.connect()

con.execute("SET temp_directory='temp_duckdb';")
con.execute("SET memory_limit='8GB';")
con.execute("SET threads=4;")
con.execute("PRAGMA disable_object_cache;")
con.execute("PRAGMA max_temp_directory_size='200GB';")

# Paths to data files
MESSAGETRACK_PARQUET = "Data/messagetrack.parquet"
MESSAGELOG_PARQUET   = "Data/messagelog2.parquet"

print("Config ready.")
```

This section initializes the DuckDB environment, sets memory and threading configurations to optimize processing of large datasets, and defines file path variables.

A.2 Data Validation and Sampling

python

```
for path in (MESSAGETRACK_PARQUET, MESSAGELOG_PARQUET):
    exists = os.path.exists(path)
    size_gb = os.path.getsize(path)/1e9 if exists else 0
    print(f"{path}: exists={exists}, size≈{size_gb:.2f} GB")


assert os.path.exists(MESSAGETRACK_PARQUET), "Missing messagetrack.parquet"
assert os.path.exists(MESSAGELOG_PARQUET), "Missing messagelog2.parquet"


print("Exploring data structure (memory-safe)...")


mt_sample = con.execute(f"SELECT * FROM '{MESSAGETRACK_PARQUET}' LIMIT
5").fetchdf()
ml_sample = con.execute(f"SELECT * FROM '{MESSAGELOG_PARQUET}' LIMIT
5").fetchdf()


print(" MessageTrack sample:")
print(mt_sample.head())
print("\n MessageLog sample:")
print(ml_sample.head())
```

Checks for file existence and sizes, then samples and prints initial rows to verify structure.

A.3 Basic Dataset Statistics

python

```
mt_basic = con.execute(f"""
  SELECT
```

```
    COUNT(*) AS total_rows,
    MIN(msgcreateddt) AS first_date,
    MAX(msgcreateddt) AS last_date
  FROM '{MESSAGETRACK_PARQUET}'
""").fetchdf()

ml_basic = con.execute(f"""
  SELECT
    COUNT(*) AS total_rows,
    MIN(auditdatetime) AS first_date,
    MAX(auditdatetime) AS last_date
  FROM '{MESSAGELOG_PARQUET}'
""").fetchdf()

print("MessageTrack:")
print(f"  • Total rows: {mt_basic['total_rows'].iloc[0]:,}")
print(f"  • Date range: {mt_basic['first_date'].iloc[0]} → {mt_basic['last_date'].iloc[0]}")

print("\nMessageLog:")
print(f"  • Total rows: {ml_basic['total_rows'].iloc[0]:,}")
print(f"  • Date range: {ml_basic['first_date'].iloc[0]} → {ml_basic['last_date'].iloc[0]}")
```

Queries basic descriptive statistics on dataset size and temporal coverage.

A.4 Current Workflow State Summary Aggregation

python

```
current_state_summary = con.execute(f"""
  SELECT
```

```
    DATE_TRUNC('month', msgcreateddt) AS creation_month,
    EXTRACT(YEAR FROM msgcreateddt) AS year,
    msgtype,
    msgstatus AS current_status,
    msgqueue  AS current_queue,
    msgdepartment,
    msgpriority,
    COUNT(*)                AS active_workflows,
    COUNT(DISTINCT msguserid) AS users_involved,
    AVG(EXTRACT(EPOCH FROM (COALESCE(msgmodifieddt, msgcreateddt) - msgcreateddt))/86400.0) AS avg_age_days,
    MIN(msgcreateddt) AS oldest_workflow,
    MAX(COALESCE(msgmodifieddt, msgcreateddt)) AS latest_activity,
    COUNT(CASE WHEN EXTRACT(EPOCH FROM (COALESCE(msgmodifieddt, msgcreateddt) - msgcreateddt))/86400.0 <= 1  THEN 1 END) AS same_day_workflows,
    COUNT(CASE WHEN EXTRACT(EPOCH FROM (COALESCE(msgmodifieddt, msgcreateddt) - msgcreateddt))/86400.0 >  30 THEN 1 END) AS long_running_workflows
  FROM '{MESSAGETRACK_PARQUET}'
  WHERE msgcreateddt IS NOT NULL
  GROUP BY 1,2,3,4,5,6,7
  ORDER BY creation_month DESC
""").fetchdf()

current_state_summary.to_csv('pbi_current_state.csv', index=False)
```

Monthly aggregation of workflow metrics including counts, average ages, and workflow categories, exported for Power BI.

A.5 Iterative Chunked Audit Log Processing

```
python
# Detect audit log date range
date_range = con.execute(f"""
  SELECT
    MIN(auditdatetime) AS start_date,
    MAX(auditdatetime) AS end_date,
    COUNT(*) AS total_records
  FROM '{MESSAGELOG_PARQUET}'
  WHERE auditdatetime IS NOT NULL
""").fetchdf()

ml_start = pd.to_datetime(date_range['start_date'].iloc[0])
ml_end   = pd.to_datetime(date_range['end_date'].iloc[0])

chunk_days = 90
log_summary_all = pd.DataFrame()
current_date = ml_start
while current_date < ml_end:
    next_date = min(current_date + timedelta(days=chunk_days), ml_end)
    df = con.execute(f"""
        SELECT
          msgid,
          COUNT(uniqueid) AS state_changes,
          MIN(auditdatetime) AS first_audit,
          MAX(auditdatetime) AS last_audit,
          COUNT(DISTINCT audituserid) AS users_touched,
```

```
        COUNT(DISTINCT auditqueue) AS queues_visited,
        COUNT(DISTINCT auditdepartment) AS departments_involved
      FROM '{MESSAGELOG_PARQUET}'
      WHERE auditdatetime >= '{current_date.strftime('%Y-%m-%d')}'
        AND auditdatetime < '{next_date.strftime('%Y-%m-%d')}'
      GROUP BY msgid
    """).fetchdf()
    log_summary_all = pd.concat([log_summary_all, df], ignore_index=True)
    current_date = next_date
```

Processes audit log data efficiently in 90-day chunks to manage memory constraints.

A.6 Quarterly Workflow Lifecycle Export

python

```
def generate_quarters(start_date: str, end_date: str):
    quarters = []
    current_start = pd.to_datetime(start_date)
    end = pd.to_datetime(end_date)
    while current_start < end:
        month = current_start.month
        if month <= 3:
            next_start = pd.Timestamp(year=current_start.year, month=4, day=1)
        elif month <= 6:
            next_start = pd.Timestamp(year=current_start.year, month=7, day=1)
        elif month <= 9:
            next_start = pd.Timestamp(year=current_start.year, month=10, day=1)
        else:
            next_start = pd.Timestamp(year=current_start.year+1, month=1, day=1)
```

```
        if next_start > end:
            next_start = end
        quarters.append((current_start.strftime("%Y-%m-%d"), next_start.strftime("%Y-%m-%d")))
        current_start = next_start
    return quarters

quarters = generate_quarters("2019-01-01", "2025-05-28")

for start_date, end_date in quarters:
    output_parquet = f"pbi_workflow_lifecycle_{start_date}_to_{end_date}.parquet"
    con.execute(f"""
      COPY (
        SELECT
          mt.msgid,
          mt.msgtype,
          mt.msgstatus AS current_status,
          mt.msgqueue AS current_queue,
          mt.msgdepartment AS current_department,
          mt.msgpriority,
          mt.msgcreateddt AS workflow_start,
          mt.msgmodifieddt AS workflow_modified,
          COALESCE(ls.state_changes, 0) AS state_changes,
          ls.first_audit,
          ls.last_audit,
          COALESCE(ls.users_touched, 0) AS users_touched,
          COALESCE(ls.queues_visited, 1) AS queues_visited,
```

```
        COALESCE(ls.departments_involved, 1) AS departments_involved,
        COALESCE(EXTRACT(EPOCH FROM (ls.last_audit - ls.first_audit))/86400.0, 0)
AS audit_span_days,
        EXTRACT(EPOCH FROM (COALESCE(mt.msgmodifieddt, mt.msgcreateddt) -
mt.msgcreateddt))/86400.0 AS processing_days,
        CASE WHEN COALESCE(ls.state_changes, 0) = 0 THEN 'No Audit Trail'
            WHEN ls.state_changes <= 3 THEN 'Simple'
            WHEN ls.state_changes <= 10 THEN 'Moderate'
            ELSE 'Complex'
        END AS complexity_category,
        CASE WHEN EXTRACT(EPOCH FROM (COALESCE(mt.msgmodifieddt,
mt.msgcreateddt) - mt.msgcreateddt))/86400.0 <= 1 THEN 'Same Day'
            WHEN EXTRACT(EPOCH FROM (COALESCE(mt.msgmodifieddt,
mt.msgcreateddt) - mt.msgcreateddt))/86400.0 <= 7 THEN 'Within Week'
            WHEN EXTRACT(EPOCH FROM (COALESCE(mt.msgmodifieddt,
mt.msgcreateddt) - mt.msgcreateddt))/86400.0 <= 30 THEN 'Within Month'
            ELSE 'Long Running'
        END AS performance_category
       FROM '{MESSAGETRACK_PARQUET}' mt
       LEFT JOIN complete_log_summary ls ON mt.msgid = ls.msgid
       WHERE mt.msgcreateddt >= DATE '{start_date}'
         AND mt.msgcreateddt < DATE '{end_date}'
     )
     TO '{output_parquet}' (FORMAT PARQUET);
   """)
```

Exports feature-rich lifecycle data partitioned quarterly for efficient dashboard consumption.