**Portfolio 1: Data Quality and Performance**

**1. Monitoring Data Stores for Performance and Availability**

Our company uses various data storage solutions to meet diverse needs across operational and analytical workloads. We use SQL databases, specifically PostgreSQL, for structured data and online transaction processing (OLTP) workloads. PostgreSQL also supports various NoSQL-like use cases, including JSON data types and extensions that enable complex graph-based queries.

We utilise cloud object-based storage solutions like Minio and SharePoint for scalable, cost-effective, and durable storage of large volumes of unstructured data. Additionally, we employ distributed file systems such as the Hadoop Distributed File System (HDFS), which is well-suited for processing and storing massive datasets across multiple nodes in a cluster.

To monitor these data stores effectively, we rely on platforms such as Dataplane and Dynatrace. Dataplane provides real-time logging and visualisation from data pipelines, helping us continuously track data flow, identify anomalies, and resolve performance issues promptly to ensure uninterrupted data processing. (*See Fig 1 below for an example).*

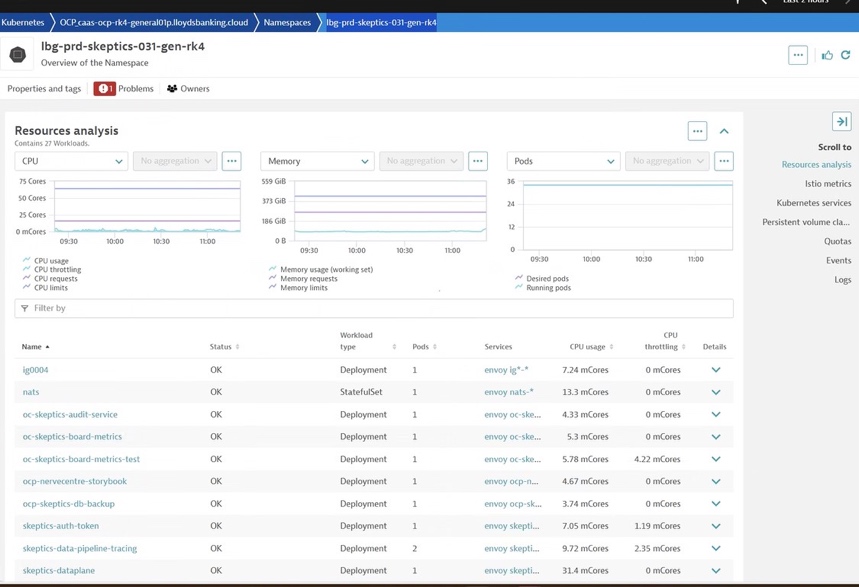
Fig 1: Dataplane screenshot showing a Pipeline Flow

A screenshot of a computer

AI-generated content may be incorrect.

Dynatrace tracks CPU and memory metrics from our OpenShift Kubernetes environment, as well as resource usage from PostgreSQL databases and Python worker nodes. This tool offers deep insights into infrastructure and application performance, helping to optimise system resources and ensure high availability by detecting and addressing potential bottlenecks proactively. As shown in Fig 2 below, Dynatrace helps us visualise and monitor critical metrics, ensuring smooth operations and maximum system efficiency.

Fig 2: Dynatrace Screenshot



**2. Data Normalisation and Its Advantages**

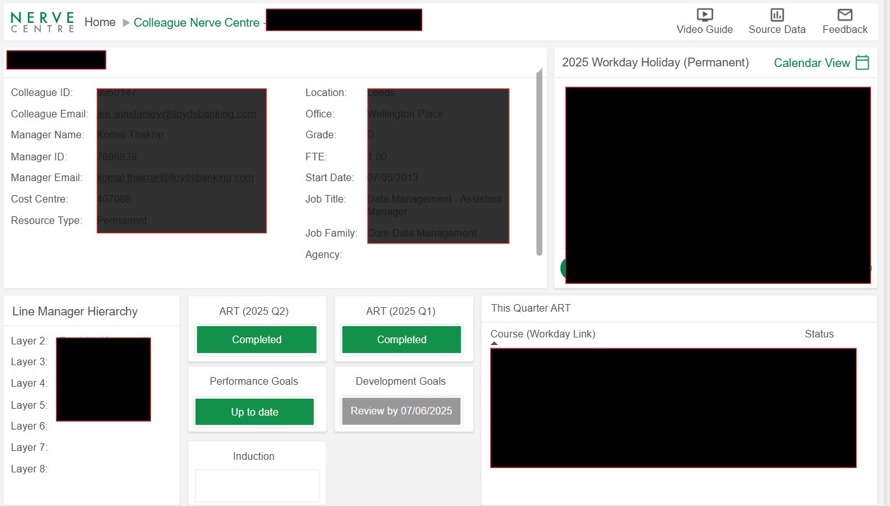
Data normalisation is a structured and systematic approach used in database design to minimise data redundancy, enhance data integrity, and ensure consistent relationships across different data entities. It involves organising data into logical tables and defining relationships through primary and foreign keys. As a data engineer, one of my key responsibilities is to ensure our databases are optimised for both performance and clarity, and normalisation plays a crucial role in achieving these objectives.

One of the fundamental advantages of data normalisation is the significant reduction of data redundancy. This means eliminating repetitive or duplicate information across tables by storing each piece of data only once and referencing it as needed. For example, instead of storing an employee’s department name in every record, we maintain a separate table for departments and use a foreign key in the employee table to establish the relationship. This results in more efficient storage, reduces the risk of inconsistencies, and makes updating information much simpler.

Another major benefit is improved data consistency. By enforcing well-defined relationships and rules between tables, we can avoid conflicting or outdated data. This is particularly important in environments where multiple applications or users’ access and update the same datasets. Enhanced data integrity is another key outcome, as normalisation encourages the use of structured schemas and validation rules, ensuring that the data remains accurate, reliable, and maintainable over time.

A practical example of this was a project I worked on with our People and Places team. The objective was to gather multiple data points related to employees to develop personalised Power BI dashboards that our colleagues could interact with. Normalising the input data allowed us to build a scalable, clean dataset that could be easily updated and reused. This not only improved the accuracy of our visualisations but also supported better decision-making. *(See Fig 3: showing Visualisation of the data.)*

Fig 3: Screenshot of one of our PowerBi Dashboard visualisations



**3. Data Quality and Risk Management**

Data quality is essential in any system that processes and manages large volumes of information, as poor-quality data can significantly impact business decisions, analytics, and operations. In my role as a data engineer, I take proactive steps to identify, assess, and mitigate risks to ensure our data pipelines are not only accurate and secure but also aligned with our strategic business objectives.

There are several common risks that can threaten data quality. Incomplete or inaccurate data often results from human error, system glitches, or limitations in the original data collection process. For example, missing values, incorrect formats, or improperly categorized entries can propagate through systems and lead to flawed insights. Another major risk is **data drift**, which occurs when the structure, meaning, or distribution of incoming data gradually changes over time. This can break data models or introduce inconsistencies in downstream applications. Integration challenges are also common when aggregating data from sources that may use varying schemas, standards, or naming conventions. Finally, non-compliant or unverified data sources can introduce legal, ethical, or regulatory concerns, particularly when handling sensitive or personal information.

To manage these risks effectively, I apply a combination of engineering practices and quality controls. Validation rules are embedded in our data pipelines to ensure inputs conform to expected formats and data types. I use data profiling tools to examine the structure, distribution, and anomalies of datasets before they move to production. Automated cleansing routines are developed to correct common issues such as typographical errors, duplicates, or missing fields. Additionally, I implement comprehensive audit trails that capture metadata, including timestamps and user actions, to provide transparency and traceability.

In one marketing data integration project, I tackled inconsistent formats and missing data from various departments. By profiling the data, applying cleansing routines, and introducing robust logging, we achieved a reliable, accurate dataset that enhanced reporting quality and supported confident decision-making.

**4. Data ingestion Frameworks and Optimisation**

In my role, we utilise workload handling software called Jira, from which we pull data for visualisation in various Power BI dashboards. Recently, we redesigned our data ingestion process and pipelines to enhance efficiency. Previously, our process involved using a custom API each morning to stream raw data from the Jira data warehouse into our cloud-based database for manipulation. This pull request involved over 150,000 rows of data in a single sequential process, which took between 38 minutes and 1 hour to complete. Despite being scheduled out of hours, this process was so CPU-intensive that it would routinely crash due to the sheer volume of the pull request, putting excess load on the server. This caused data to be missed, and the process had to be restarted. *(Fig 4 shows the previous data plane pipeline process and time taken).*

To address these issues, we redesigned our process to take advantage of micro-batching and multi-processing. The new process runs every hour during the day using micro-batching and a larger report after hours using multi-processing.

The evening run utilises 15 parallel CPU processing tasks, each pulling approximately 10,000 items. This process sets the baseline of changes in the workload for that day and now takes less than 5 minutes to complete *(see Fig 5).*

Additionally, we designed an hourly pull throughout the day using micro-batching that only pulls through changes in the data. This process also now only takes around 5 minutes to complete (*see Fig 6).*

As described above, this new process has significantly improved the durability of the data pull. By using micro-batching and obtaining updates when needed, the process has become highly scalable with minimal impact on performance. The CPU power is distributed in a far more efficient manner, ensuring that the system remains stable and responsive.

In summary, the redesign of our data ingestion framework has led to substantial improvements in efficiency, reliability, and scalability. The implementation of micro-batching and multi-processing has not only reduced the time required for data ingestion but also minimised the risk of system crashes and data loss. This optimised process ensures that our data is consistently up-to-date and readily available for visualisation and analysis in Power BI dashboards, supporting better decision-making and operational efficiency.

Fig 4: Dataplane screenshot of old process

A screenshot of a computer

AI-generated content may be incorrect.

Fig 5 New nightly process

A screenshot of a computer

AI-generated content may be incorrect.

Fig 6: New hourly Process

A screenshot of a computer

AI-generated content may be incorrect.

**Extension: Descriptive, Predictive and Prescriptive Analytics**

Analytics helps organisations make sense of their data and use it to guide decision-making. As a data engineer, my role is to ensure that the infrastructure supporting analytics is reliable, scalable, and provides clean, structured data.

**Descriptive analytics** focuses on analysing historical data to understand what has already happened. It typically involves aggregating data, generating dashboards, and creating visual reports that reveal patterns and trends. For example, we built a dashboard that pulls together all employee training records, holiday bookings and various other datapoints into one visualisation *(See Fig 3 from part 2 of this document).*

**Predictive analytics** goes further by using statistical models and machine learning algorithms to analyse current and historical data to forecast future outcomes. By identifying patterns and trends, it enables us to make informed decisions about likely future events, such as customer behaviour, market shifts, or operational risks.

**Prescriptive analytics** builds on predictive insights by recommending specific actions to take, enabling banks to move from insight to action with greater confidence. It often uses decision trees, optimisation techniques, and scenario analysis to evaluate options. We could use prescriptive tools to suggest optimal loan approval strategies based on customer risk profiles, tailor product offerings to maximise profitability, or allocate resources more efficiently across branches and service channels.

Each of these analytics types play a unique role. Descriptive analytics answers “What happened?”, predictive analytics answers “What is likely to happen?” and prescriptive analytics answers “What should we do about it?”