PAYROLL DATA PROCESSING AND VISUALIZATION

Abstract

This project focuses on developing an integrated payroll management framework that leverages SQL for data processing and Power BI for visualization. Payroll data, being one of the most sensitive and critical components of any organization, requires accurate handling to ensure compliance, transparency and efficiency. By creating a streamlined pipeline from raw payroll records to insightful dashboards, this project aims to enhance decision-making, detect anomalies such as potential wage underpayments and support compliance with evolving employment regulations. The solution will ultimately provide organizations with a robust, user-friendly platform to manage payroll data and gain actionable insights.

Nhat Minh Dang

Contents

A) Introduction:

Payroll is at the heart of employee satisfaction, legal compliance, and organizational accountability. With the increasing importance of accurate payroll processing—particularly considering stricter wage compliance regulations—businesses must adopt data-driven approaches to ensure reliability and transparency. This project addresses that need by building a system where payroll data is first processed using SQL to ensure data integrity, consistency, and aggregation and then visualized using Power BI to present trends, insights and key performance indicators (KPIs). The interactive dashboards will allow managers and HR professionals to explore payroll distributions, track expenses, and identify compliance risks in real time. This project not only demonstrates the technical workflow from database management to business intelligence but also highlights the role of analytics in strengthening trust and accountability within payroll systems.

B) Data Overview and Processing through SQL Server:

1) Define the Objective:

Overall, this Power BI dashboard to ensure payroll accuracy, monitor over and under payments, followed by evaluating employee compensation or benefits.

These are the key questions that I will answer:

- How much is being overpaid or underpaid over time?
- Are certain employee groups (job type, contract type) more prone to payroll issues?
- What patterns exist in overtime, undertime and leave usage?
- How does payroll compliance affect employee status and contract renewals?

2) Datasets:

The datasets includes 14 files that are in .csv format:

- Allowance: Extra payments made to employees within specific periods. There are two main types: Laundry Allowance and First Aid Allowance.
- Bonus: Extra rewards for employees within specific events such as Retention Bonus, Christmas Bonus, Performance bonus, etc.
- Combined holiday: List of holiday plans that were implemented. For example, if the Christmas falls on Sunday, there will be extra day-offs for the following days as additional holidays.
- Contract details: Contains information about the employees such as job type, title, active status, pay rate, etc.
- Date: A dataset created to help retrieving the chronological time.
- Employee details: List of employees along with their personal information.
- Employee leaves: List of leaves recorded.
- Junior pay rates: Wages for employees within specific age group.

- Minimum pay rates: The minimum wages implemented by company throughout specific periods.
- Payrate adjustment: Wage modifications for specific events such as overtime, nightshift, penalties, etc.
- Roster: Employee's work schedule recorded.
- Tax rates: Tax calculation formula within each specific year.
- Time off in lieu: Overtime events recorded and usage.
- Timesheet: Include transactions information.

3) Loading Data into SQL server:

The datasets were loaded into SQL server and managed by using SSMS. These are my standard data types when loading datasets:

- Any ID detail was defined as nvarchar(50).
- Any number of hours or pay rate with decimal part will be defined as decimal(18, 10). Otherwise, it was attached with int or tinyint (if small).

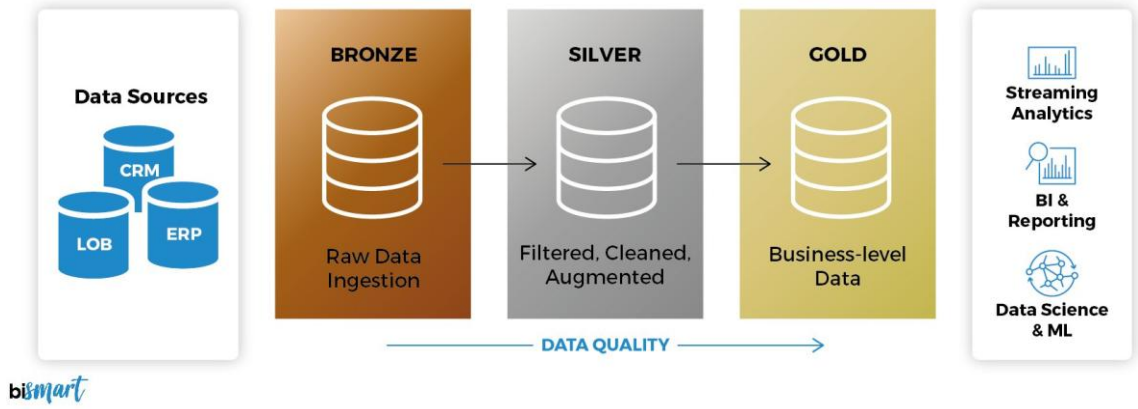

Figure 1: Medallion Architecture.

In this project, I will use three data schemas: Landing, Staging and Mart. These schemas represent three steps in Medallion Architecture. Medallion architecture is a powerful data design pattern used in modern Lakehouse systems to progressively refine and organize data across three distinct layers—Bronze, Silver and Gold—each representing a step in data quality and usability. At its foundation, the Bronze layer ingests raw, unprocessed data from diverse sources such as CRM, ERP or LOB systems, preserving its original form for traceability and historical analysis. This raw data then flows into the Silver layer, where it undergoes cleaning, validation and enrichment—removing duplicates, standardizing formats, and integrating disparate datasets to create a more coherent and reliable view. Finally, the Gold layer transforms this refined data into business-ready assets through dimensional modelling, aggregation and domain-specific enhancements, making it ideal for executive dashboards, machine learning models and advanced analytics. This tiered approach not only ensures data integrity and governance but also enables modular development, scalability and efficient collaboration across data engineering, analytics, and decision-making teams. By separating

concerns and incrementally improving data quality, medallion architecture empowers organizations to build robust, flexible pipelines that support both operational reporting and strategic insights.

Because the date table was created individually, it was put into the Mart layer. For other files, they were imported into Landing layer.

4) Processing to Staging Layer:

In the first part of the SQL code, I transformed the data from landing into cleaner, structured datasets in the Staging Layer. These are the chosen data columns within each file:

- Allowance: allowance_id, employee_id, allowance_type, allowance_amount, allowance_start_date, allowance_end_date.
- Bonusses: bonus_id, employee_id, bonus_type, bonus_amount, bonus_date.
- Contracts: contract_id, employee_id, start_date, end_date, pay_rate, job_title, payment_frequency, contract_type, employment_type, contract_status.
- Leaves: leave_id, employee_id, leave_type, leave_start_date, leave_end_date, leave_hours.
- Employees: employee_id, employee_first_name, employee_middle_name, employee_last_name, employee_gender, employee_location, date_of_birth, hire_date, termination_date, employee_status.
- Junior pay rate: Map age brackets to custom pay_rate_id codes and select age, percent_of_adult_pay_rate.
- Minimum pay rates: pay_rate_id, effect_from, effect_to, hourly_permanent_rate, hourly_casual_rate
- Pay rate adjustment: Clean and standardized rate type and pay rate id using replace and upper and select description, rate_calculation
- Roster: roster_id, employee_id, shift, hours, work_date, services, location, pay_period_id
- Tax rate: Constructs a unique tax_rate_id using cleaned year and income ranges.
- Time off in lieu: toil_id, employee_id, overtime_date, toil_hours_accrued, toil_usage_date
- Timesheet: timesheet_id, employee_code AS employee_id, timesheet_transaction_date, start_time, end_time, timesheet_transaction_hours, pay_period_id; rename employee code to employee id.

5) Marts Layer Finalization:

From the staging layer, I implement the Gold layer to create dimensional and fact for analytics, reporting and modelling. These are the dimension and fact tables:

a. Dimension view:

- Dim Contracts: Using contract_id and start_date and generate contract_pk with SHA1.
- Dim Employees: Create employee_pk is hashed from employee_id and hire_date.
- Dim junior pay rates: Converts percentage to multiplier for easier wage calculation.
- Dim minimum pay rates: No changes was made.

- Dim pay period: Aggregates dates into monthly, fortnightly, and weekly periods with labels and payday logic.
- Dim pay rate adjustments: Adds rate_multiplier logic for common pay scenarios (for example from overtime_rate_after_2_hrs into 2.0)
- Dim tax rate: No change was made.
- Dim date: Remain the same.

b. Fact views:

- Fact allowances:
    - Join dim employees via employee_id.
    - Join dim_contracts via employee_id and date range.
    - Join dim_dates via allowance_start_date.
    - Maps allowance_start_date and end_date to surrogate date keys.
    - Derives pay_period_fk based on contract's payment_frequency.
- Fact bonusses and Fact employee leaves: Similar as allowance.
- Fact roster:
    - Join dim_employees for employee surrogate key employee_fk.
    - Join dim_contracts to contextualize the shift within the correct contract.
    - Join dim_dates to derive pay_period_fk based on work_date.
- Fact time off in lieu and timesheet: Same structure as roster.

C) Visualization through Power BI:

1) Importing data and matching relationships:

Using datasets from SQL, I import the files into Power BI for visualization, followed by naming proper dimension and fact tables. Below is my relationship diagram (further details about relationship can be viewed in the power bi report):

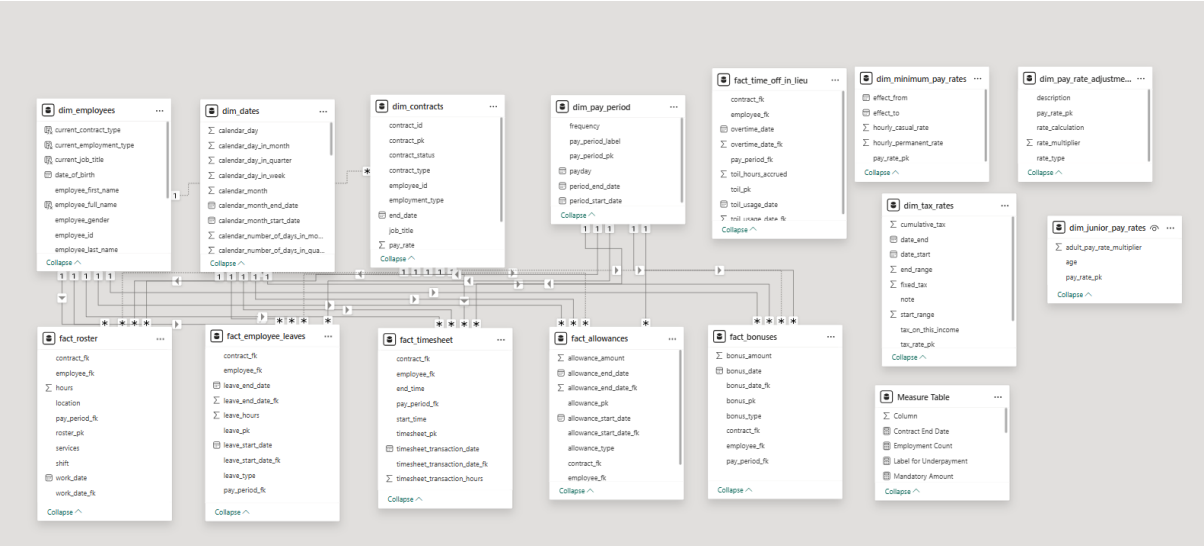

Figure 2: Relationship Diagram.

2) Result Report:

a. Payroll Refinement:

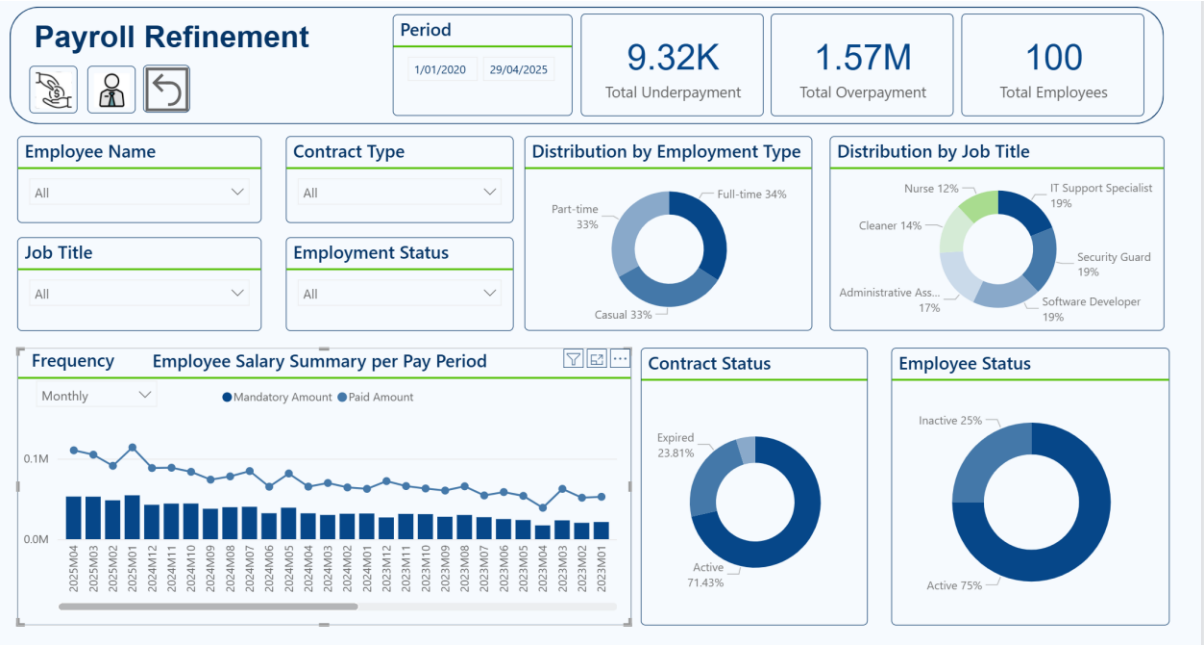

Figure 3: Payroll Dashboard 1.

Overall, among 100 employees, the distribution between Part time, Full time and Casual Employees were balanced at 33%. Therefore, the risk did not fall heavily into any group. However, the job title distribution depended on IT Support Specialist, Security guard and Software Developer with 19%. These are the positions that should be focused on the payroll refinement process. Furthermore, in the employee and contract status, the active employees still contributed in major. This indicated that the company still attracts employees for long term with appropriate policies. There was a portion of 24% for expired and 6% for terminated. These figures might explain some payroll mismatches. While there were 34.8k Underpayment, the huge amount of 5.39M Overpayment showed the bigger risk.

Over the period from 2021 to April 2025, there was a significant increase in both Mandatory amount and Paid amount. However, the difference between Mandatory amount and Paid amount growth remarkably since September 2022. From 2022 to 2024, while the Paid amount fluctuated between 0.06M to 0.08M, the Mandatory amount went ups and down between 0.04M and 0.05M. This was because the huge impact of Covid pandemic that slowed down the development of companies. Only after June 2024 where the Paid amount increased significantly to more than 0.1M. However, there was only a slight increase of Mandatory amount. This indicated that the company should cut down the overpayment to maintain the profit overtime.

Figure 4: Mandatory and Paid amount for Part time, Casual and Full time.

When observing into each contract type, it can be seen that the Casual reached the highest among those three with 50k. Surprisingly, full time positions contributed the lowest amount with only 20k. During the period, while the Part time and Casual went up gradually and remained stable from 2023 onward, the demand for Full time positions got concentrated after 2025 for the paid amount. In contrast, instead of increasing slowly like part time and casual, the full-time positions witnessed ups and downs slightly between 20k. This demonstrated a good sign of preserving talented employees of the full-time type and only invest into more workers during peak periods only. However, this also alerted the company about huge dependent on part time and casual staffs, which result in unstable performance when the job market does not satisfy the demand of company.

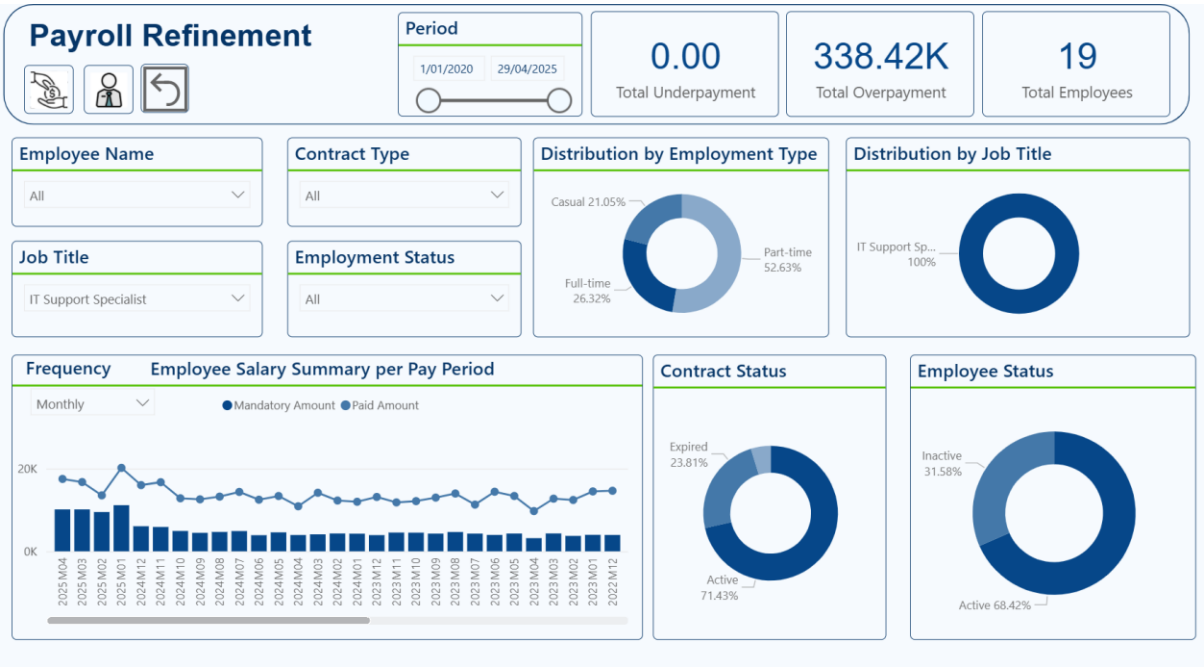

Figure 5: IT Support Specialist.

In the IT Support Specialist, Part time positions showed the most contribution, followed by casual and full time. Furthermore, this position only growth in demand from 2025 onward. This demonstrated that although taking a huge space in employment type, IT Support Specialist was not a long-term positions that the company focus on.

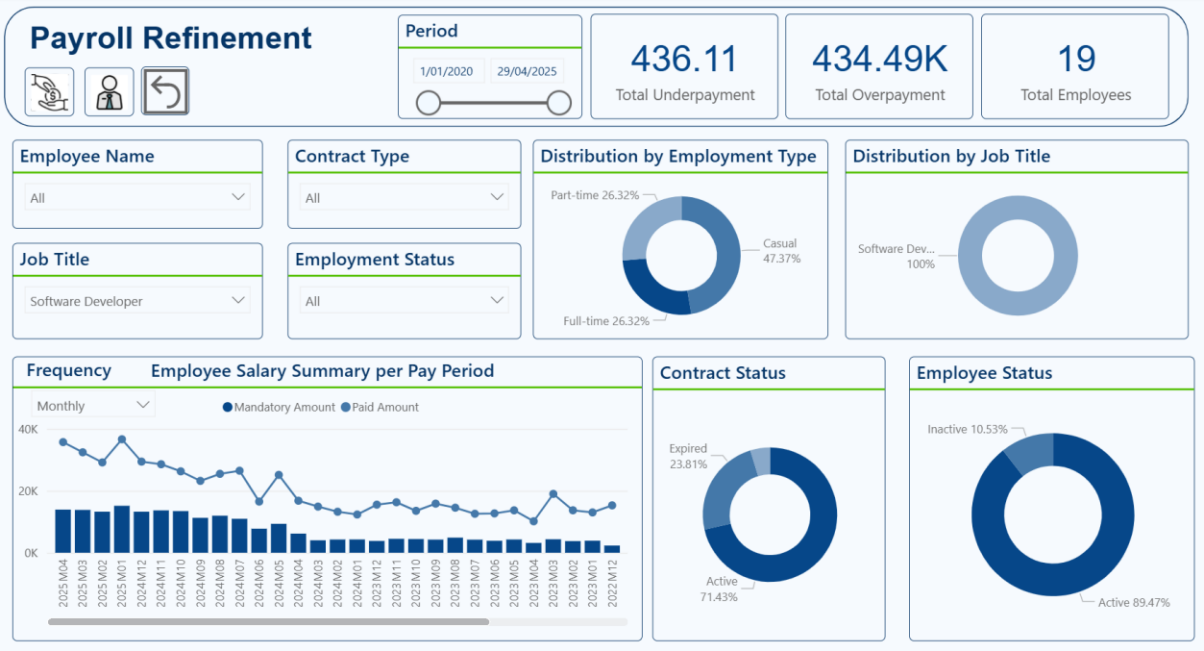

Figure 6: Software Developer.

However, when comparing to Software developer, although the need for this position rose up from the middle of 2024, the proportion of casual positions took up nearly half of the ratio. However, the paid and mandatory amount was significantly higher than IT Support Specialist. The result had shown that although Software Developer was more valuable than IT Support Specialist but less lasting.

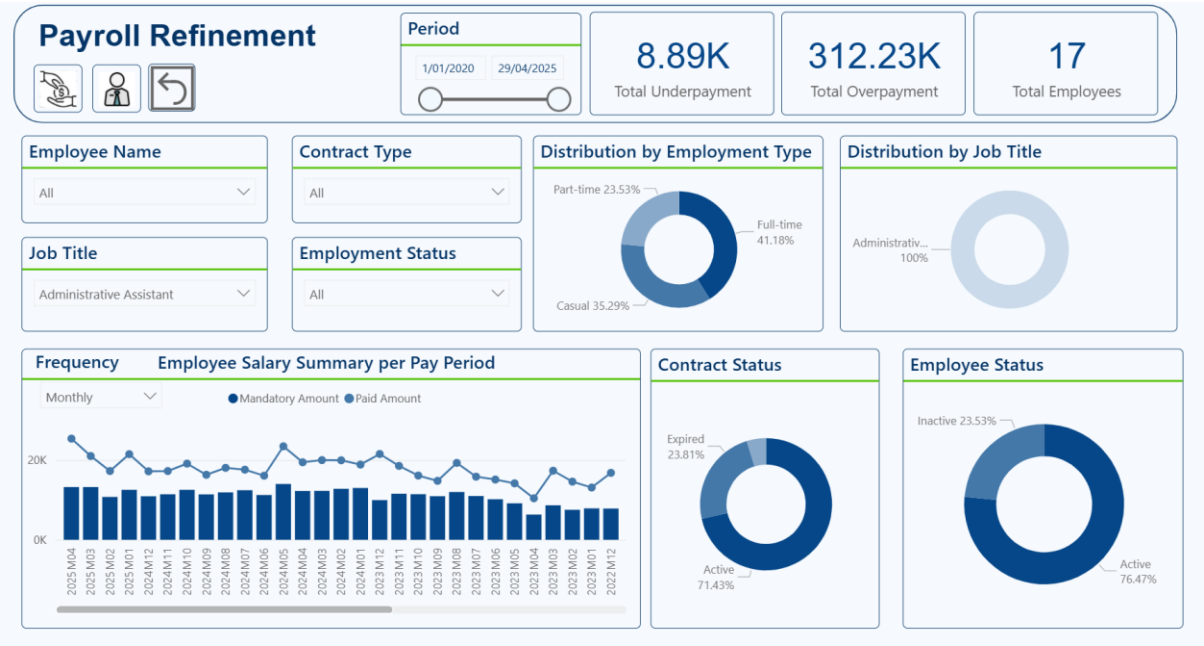

Figure 7: Administrative Assistant.

Moving on to the Administrative Assistant, unlike other positions, full time employees took account the most in the company. Furthermore, there was a slight fluctuation in both Mandatory amount and Paid amount. Therefore, it can be concluded that Administrative

Assistant was almost irreplaceable part of the company. However, after digesting deeper into those most contributed positions, I suggested that the

b. Employee Analysis:

In the second dashboard, I digested deeper into each employee to help the users keep tracking the progress of each worker. Below are few examples of some employees:

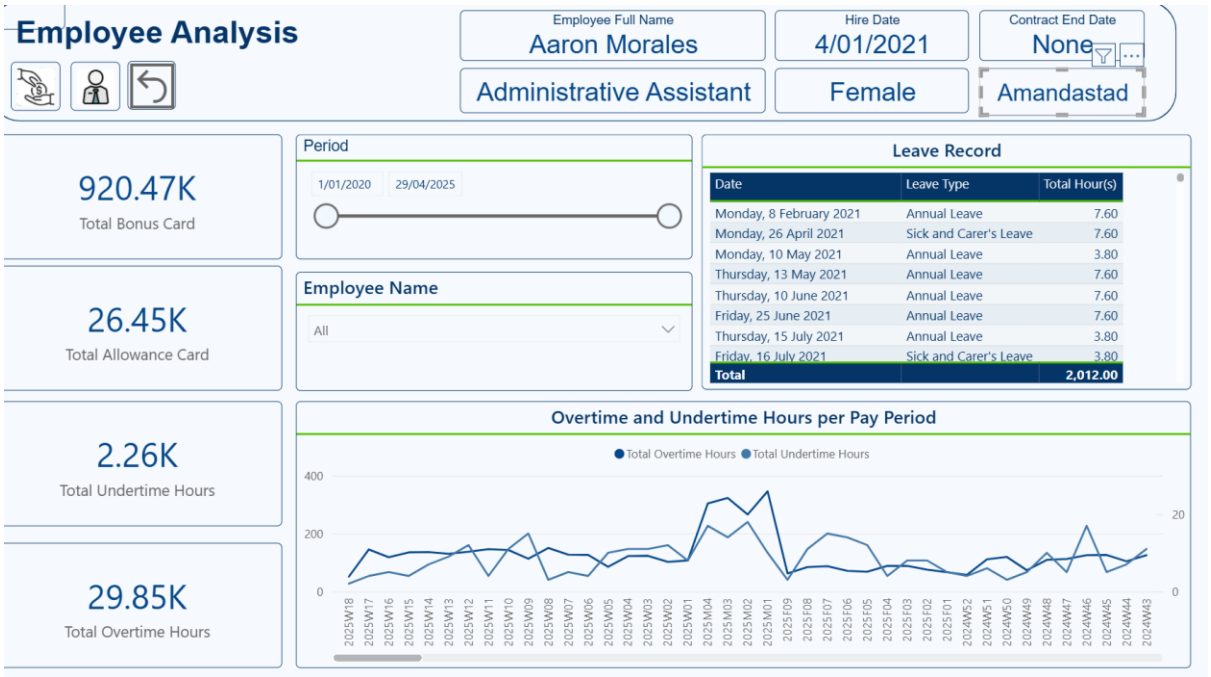

Figure 8: Overall Report.

As can be seen from the dashboard, for 5 years, there were nearly 1 million bonus cards and approximately 26.5k allowance cards. Notably, in contrast of 2.26k undertime hours, there were almost 30k overtime hours. This demonstrated that although the company has a huge amount of overtime hours for demand, the managers also offer extra rewards and leaves to ensure the balance between work and other daily activities of the employees. Furthermore, there was only a dramatically witness in the first four months of 2025 where the need of overtime employees reached a peak of 390 hours. In addition, there was a close connection between overtime and undertime hours, showing that the company still maintained the overall working hours for her.

Figure 9: Aaron Morales Report.

For Aaron Morales, she was hired in 2021 and lived in North Kelly. As can be seen from the graph, her overtime hours fluctuated between 20 to 40 hours while the undertime was between 1 to 3 hours only. She mostly took leaves every 2 or 3 months. This shows that she had a balanced life between work and personal plans and a dedicated staff.

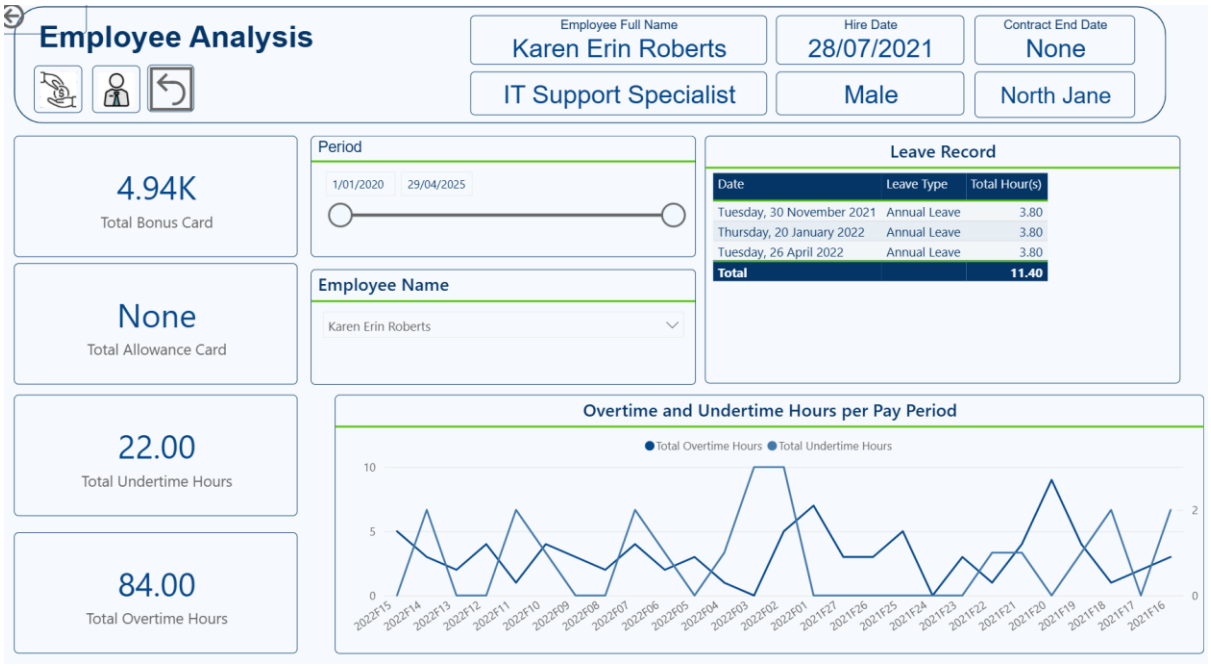

Figure 10: Roberts Report.

Moving on to Rober, unlike Aaron, although he was hired in 2021, his overtime and undertime hours was massively lower than her with only 84 hours and 22 hours only. Furthermore, the difference between overtime and undertime hours was larger than Aaron with a peak of 10 hours. Therefore, it can be concluded that there was an enormous distinction between the workload of some positions, especially IT Support Specialist and Aaron Morales.

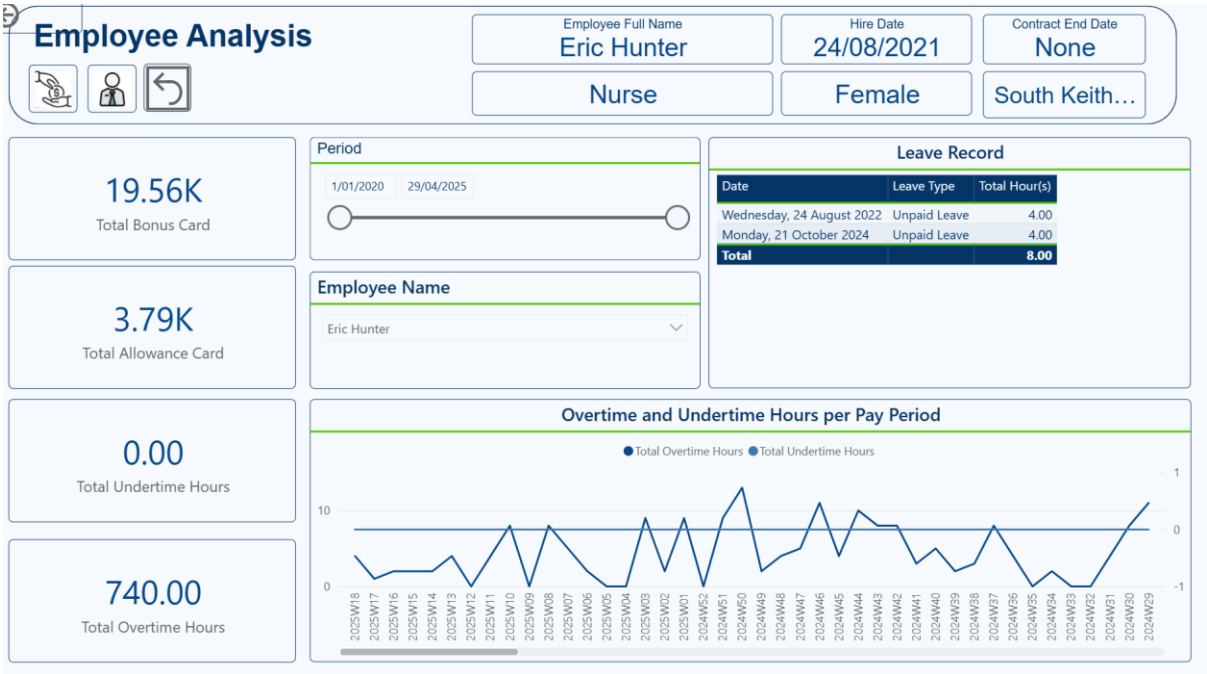

Figure 11: Eric Report.

In some minor positions, Eric has shown some remarkable details. While the overtime hours were nearly 740 hours, she has none undertime hour. Furthermore, she has only 2 leave events for the whole period. This suggested that the company should concentrate more on these hard-working staffs to ensure a healthy environment for both physical and mental.

D) Conclusions:

Overall, the salary payment process was quite balance between each contract type. However, the heavy dependence on short term employees for major positions should be considered to increase the full-time employee. There was also a high risk because of huge overpayment that was needed to be cut down to ensure overall profit. When going deeper into each staff, I found that while some employees got extra care from the company, others devoted staffs should get more concentration to ensure harmony working environment.