Creating a Dashboard for Clockster

Project Assignment 2 Batch 4 Group 8

Team Leader:	Olay, Aaron (Cohort 56)
Team Members**:**	Bernardino, Edilou (Cohort 139-A)
	Cojuangco, Leomar H. (Cohort 147-B)
	Ebreo, Vanessa (Cohort 56)
	Libunao, Rob (Cohort 139-A)

PROBLEM STATEMENT

As a team of data analytics for a medical company (Clockster), we have been provided by the HR department with data that includes 10-15 parameters per day per year on arrival/departure time, vacations, sick days, time off, etc. on 1000+ employees of the company.

Our main objective is to answer the questions which the CEO has formulated. There are 3 questions forming our main objectives. The 3rd question being comprised of 2 sub-questions. Each main objective / question should have its own dashboard in Power BI.

To achieve this, we must perform data cleaning and transformation, data analysis and visualization and complete the project within 3 weeks. Our deliverables are a PDF report, the Power BI dashboards, a workspace documentation and a 10-15 minute video presentation.

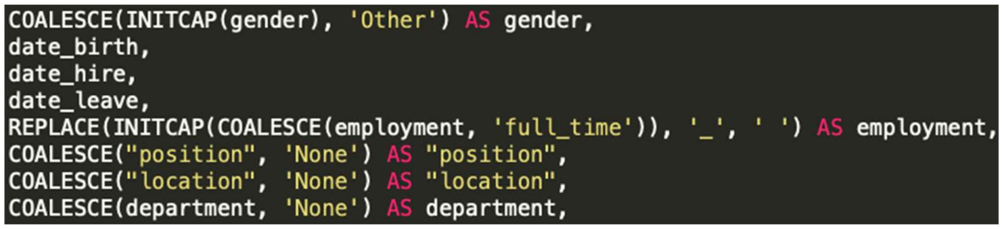
METHODOLOGY

Data Cleaning and Transformation (SQL)

The team utilized PostgreSQL at this stage to import, clean, and convert the supplied data into a dataset appropriate for loading into Power BI. The SQL script used is included.

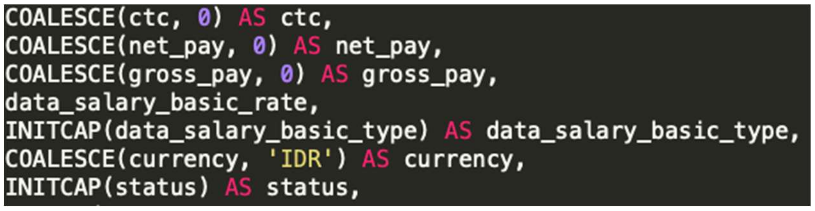
The fields of the original/raw datasets served as the basis for the initial creation and mapping of database tables. The necessary tables were then loaded with each CSV dataset file, ready for validation and transformation.

The datasets for users and payroll just needed simple blank/null value substitutions, removals of empty columns (first name and last name), as well as duplicate rows.



```
COALESCE(INITCAP(gender), 'Other') AS gender,
date_birth,
date_hire,
date_leave,
REPLACE(INITCAP(COALESCE(employment, 'full_time')), '_', ' ') AS employment,
COALESCE("position", 'None') AS "position",
COALESCE("location", 'None') AS "location",
COALESCE(department, 'None') AS department,
```

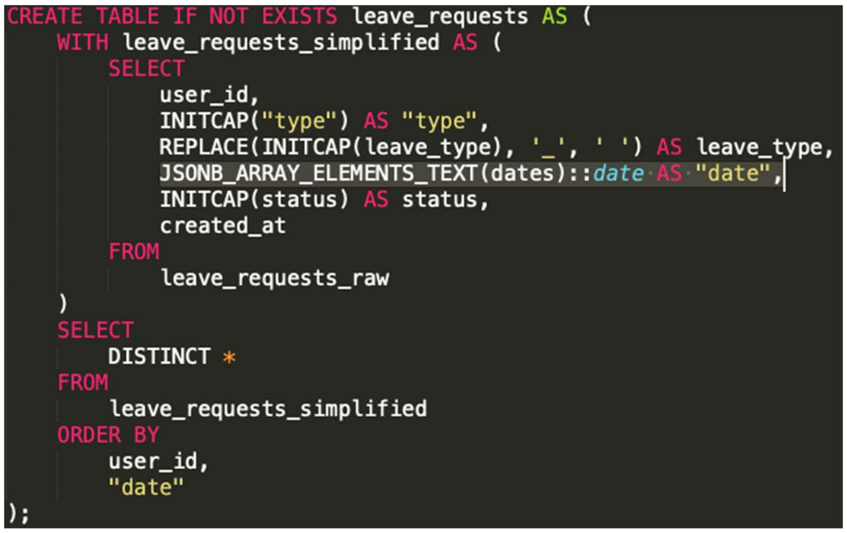
Fig. 1: Snippet of users SQL



```
COALESCE(ctc, 0) AS ctc,
COALESCE(net_pay, 0) AS net_pay,
COALESCE(gross_pay, 0) AS gross_pay,
data_salary_basic_rate,
INITCAP(data_salary_basic_type) AS data_salary_basic_type,
COALESCE(currency, 'IDR') AS currency,
INITCAP(status) AS status,
```

Fig. 2: Snippet of payroll SQL

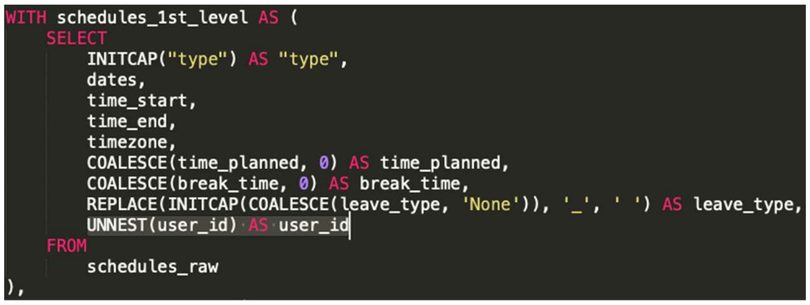
For the leave_requests dataset, apart from the similar cleaning process done with users and payroll, there was also an additional step for the list of dates to be split into rows. We were able to achieve this using the JSONB_ARRAY_ELEMENTS_TEXT function to match the JSONB data type used to store the date field.



```
CREATE TABLE IF NOT EXISTS leave_requests AS (
    WITH leave_requests_simplified AS (
        SELECT
            user_id,
            INITCAP("type") AS "type",
            REPLACE(INITCAP(leave_type), '_', ' ') AS leave_type,
            JSONB_ARRAY_ELEMENTS_TEXT(dates)::date AS "date",
            INITCAP(status) AS status,
            created_at
        FROM
            leave_requests_raw
    )
    SELECT
        DISTINCT *
    FROM
        leave_requests_simplified
    ORDER BY
        user_id,
        "date"
);
```

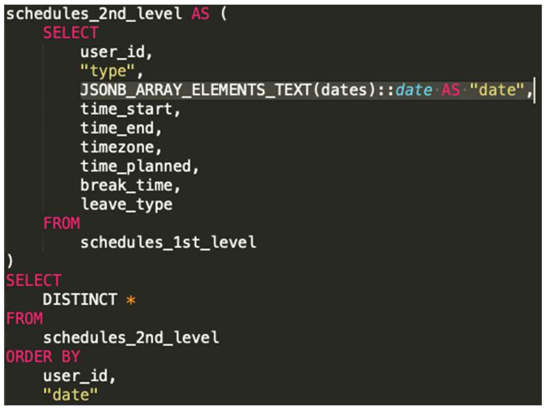
Fig. 3: Snippet of leave_requests SQL

A similar approach was used for the schedules dataset as with leave_requests, but because there are two fields with list values (user_id and date), it was necessary to split the values into rows first by user_id and then by date. This is done to prevent unnecessary data duplication and guarantee that the original data for each field is preserved throughout the procedure.



```sql
WITH schedules_1st_level AS (
    SELECT
        INITCAP("type") AS "type",
        dates,
        time_start,
        time_end,
        timezone,
        COALESCE(time_planned, 0) AS time_planned,
        COALESCE(break_time, 0) AS break_time,
        REPLACE(INITCAP(COALESCE(leave_type, 'None')), '_', ' ') AS leave_type,
        UNNEST(user_id) AS user_id
    FROM
        schedules_raw
),
```

Fig. 4a: Snippet of schedules SQL (user_id - split into rows)

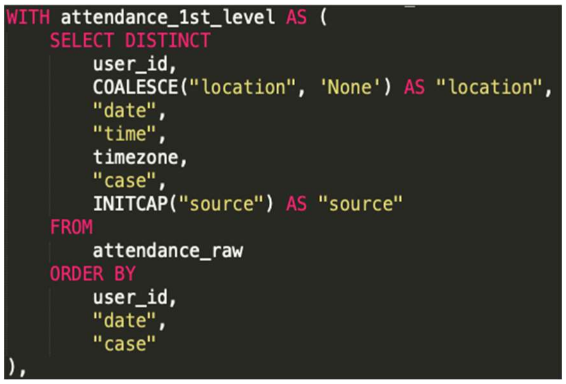


```sql
schedules_2nd_level AS (
    SELECT
        user_id,
        "type",
        JSONB_ARRAY_ELEMENTS_TEXT(dates)::date AS "date",
        time_start,
        time_end,
        timezone,
        time_planned,
        break_time,
        leave_type
    FROM
        schedules_1st_level
)
SELECT
    DISTINCT *
FROM
    schedules_2nd_level
ORDER BY
    user_id,
    "date"
```

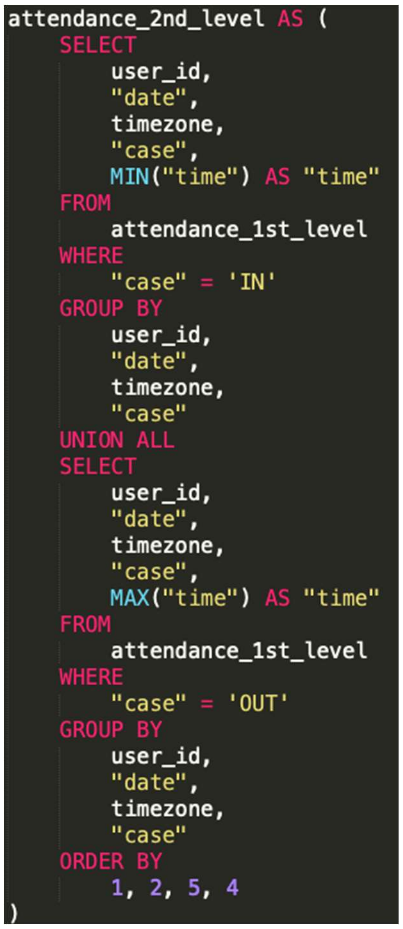
Fig. 4b: Snippet of schedules SQL (dates - split into rows)

For the attendance dataset, it was seen that there are multiple instances of IN cases falling on the same log date, and similarly, multiple instances of OUT cases. The team thought about taking the earliest IN time and the latest OUT time as part of deleting the duplicates. To execute this, the following steps were taken:

1. The dataset was split into two groups: the IN instances were in one, while the OUT cases were in the other. The location and source were previously excluded from the grouping procedure because they produced various combinations that prevented the grouping from fully extracting the earliest IN and latest OUT. Moreover, BREAK instances were excluded since they are shift-related (between IN and OUT times).
2. Both groups were combined using UNION to produce the transformed dataset.



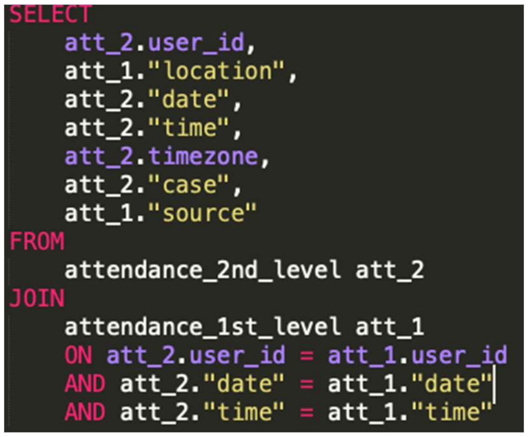
```
WITH attendance_1st_level AS (
    SELECT DISTINCT
        user_id,
        COALESCE("location", 'None') AS "location",
        "date",
        "time",
        timezone,
        "case",
        INITCAP("source") AS "source"
    FROM
        attendance_raw
    ORDER BY
        user_id,
        "date",
        "case"
),
```



```
attendance_2nd_level AS (
    SELECT
        user_id,
        "date",
        timezone,
        "case",
        MIN("time") AS "time"
    FROM
        attendance_1st_level
    WHERE
        "case" = 'IN'
    GROUP BY
        user_id,
        "date",
        timezone,
        "case"
    UNION ALL
    SELECT
        user_id,
        "date",
        timezone,
        "case",
        MAX("time") AS "time"
    FROM
        attendance_1st_level
    WHERE
        "case" = 'OUT'
    GROUP BY
        user_id,
        "date",
        timezone,
        "case"
    ORDER BY
        1, 2, 5, 4
)
```

Fig. 5a: Snippet of attendance SQL (cleaning, grouping, and union parts)

3. The transformed result set was joined with the dataset that existed before the grouping to recover the matching location and source information.



```
SELECT
    att_2.user_id,
    att_1."location",
    att_2."date",
    att_2."time",
    att_2.timezone,
    att_2."case",
    att_1."source"
FROM
    attendance_2nd_level att_2
JOIN
    attendance_1st_level att_1
    ON att_2.user_id = att_1.user_id
    AND att_2."date" = att_1."date"
    AND att_2."time" = att_1."time"
```

Fig. 5b: Snippet of attendance SQL (join part)

Data Cleaning and Transformation (PowerBI)

After cleaning and transforming the datasets in SQL, the following were made to address the various objectives of the case.

1. All cleaned and updated five (5) csv files were loaded in Power BI and the following tables were created: Attendance, Leave Requests, Payroll, Schedules, Users.
2. Under "Schedules" table> 'Type' column, other types were filtered out leaving only work schedules in the table.
3. Then using merge queries, "Schedules" table was merged with "Attendance" table using left outer join with matching columns under 'User ID' and 'Date'. As a result, columns 'Time' and 'Case' from "Attendance" table were added to "Schedules" table.
4. After merging the tables, null values were filtered out under 'Case' column.
5. To calculate the number of minutes that employees were late, a custom column 'Minutes Late' was added using power query.
   Syntax Used: *Duration.TotalMinutes([time] - [time_start])*
6. For the next step, we added again a custom column 'Punctuality Check' using power query to check if an employee was tardy or punctual for every time-in / time-out.
   Syntax Used: *if (([Minutes Late] >= 10 and [case] = "IN") or ([time] < [time_end] and* [case] = "OUT")) then "Tardy" else "Punctual"
   At this point, it should be noted that what would be considered 'outliers' (Minutes Late > 2 hours) were kept and not filtered out. These rows were significant in number, and to avoid losing other important data along with them, it was decided to keep the outliers in.
7. Then, another custom column 'Tardiness Check' was added using power query to determine if an employee's tardiness was due to being late or leaving early.
   Syntax Used: *if [Punctuality Check] = "Tardy" and [case] = "IN" then "Late" else if [Punctuality Check] = "Tardy" and [case] = "OUT" then "Left Early" else "Not Tardy"*
8. To get the day of the week for every logged attendance, custom column 'Day of the Week' was added using power query.
   Syntax Used: *Date.DayOfWeekName([date])*
9. To get the month for every logged attendance, custom column 'Month Name' was added using power query.
   Syntax Used: *Date.MonthName([date])*
10. Using DAX, the following measures were made to help with the data analysis:
    a. Late = COUNTROWS (FILTER(Schedules, Schedules[Tardiness Check] = "Late"))
    b. Left Early = COUNTROWS(FILTER(Schedules, Schedules[Tardiness Check] = "Left Early"))
    c. Punctual = COUNTROWS(FILTER(Schedules, Schedules[Punctuality Check] = "Punctual"))
    d. Tardy = COUNTROWS(FILTER(Schedules, Schedules[Punctuality Check] = "Tardy"))
11. A home table named 'Objective Measures' was created to contain all the calculated measures as shown in Figure 6 below.

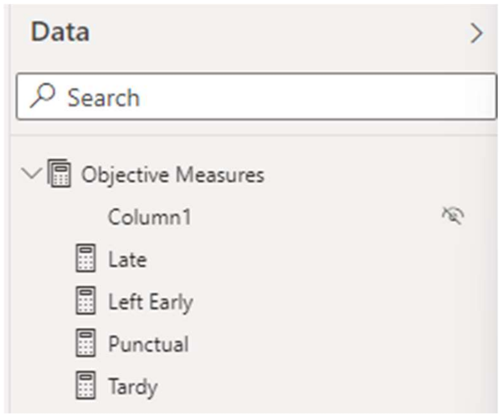


Figure 6. Objective Measures Table

Data Exploration

Following the establishment of Objective Measures, we begin data exploration by modeling the scheme as shown in Figure 7 and analyzing each business objective to create the data visualization.

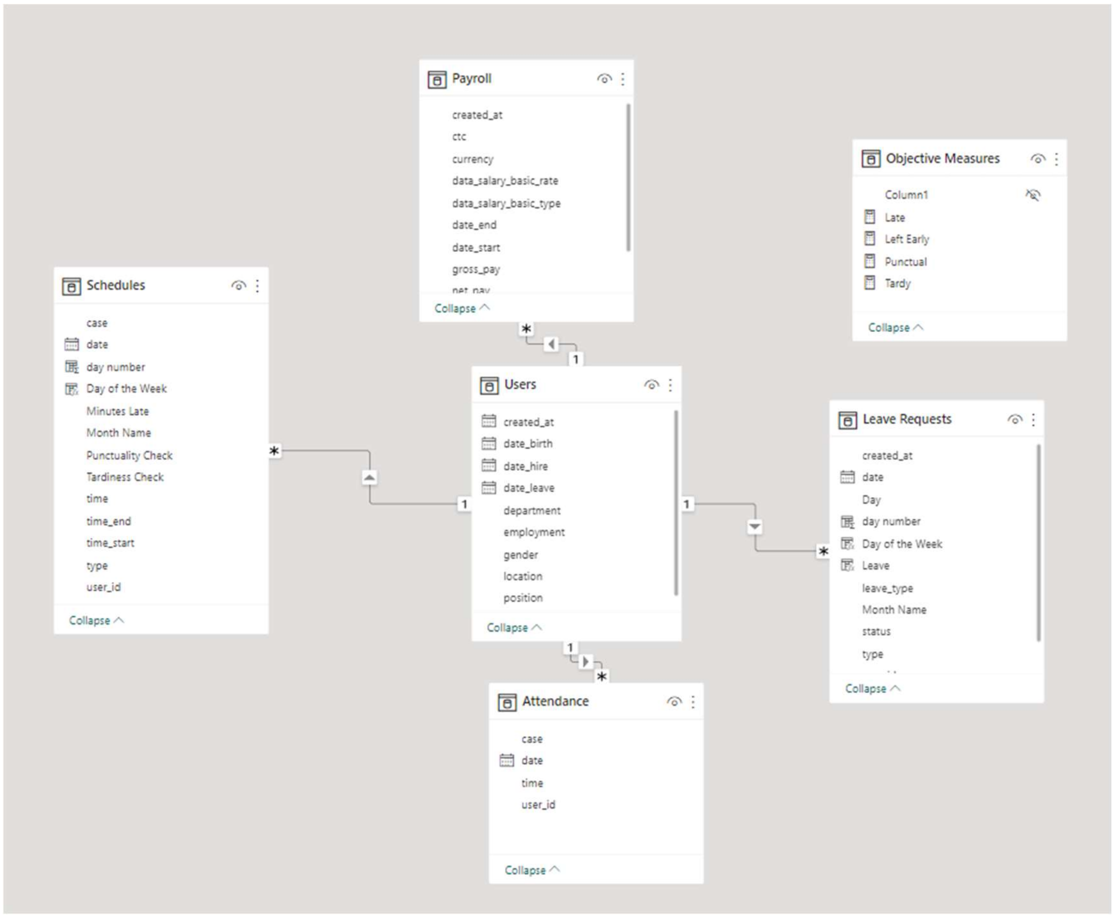
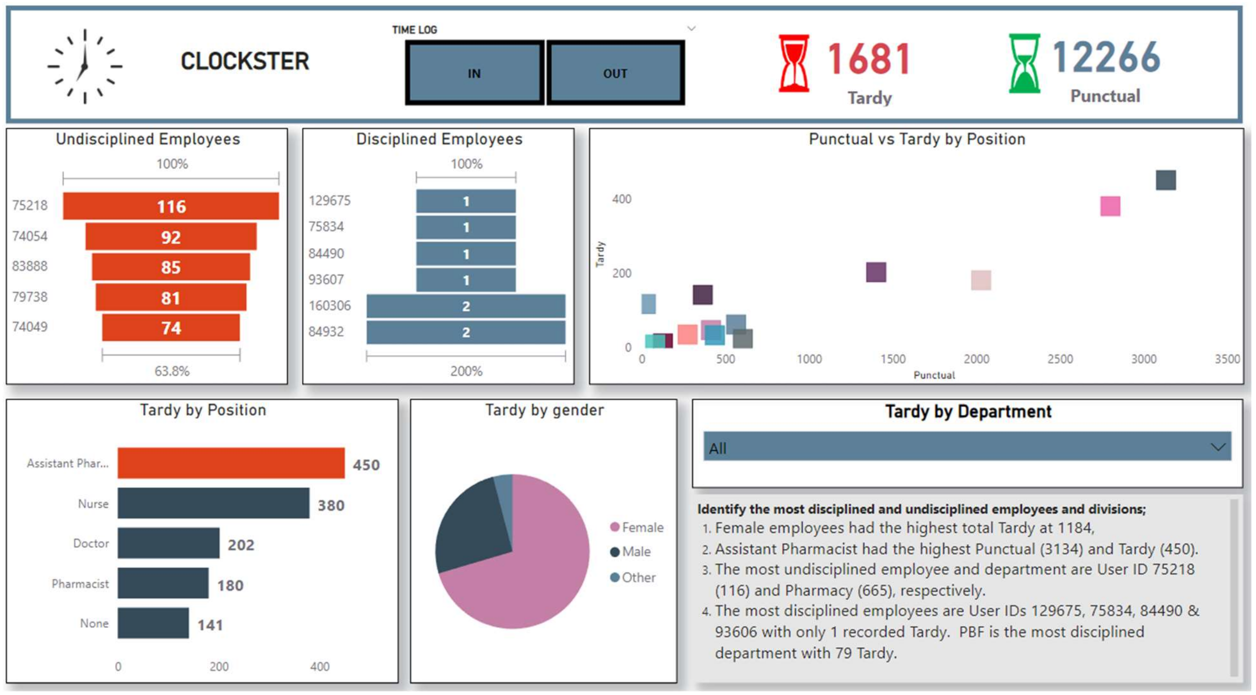


Figure 7. Modeled Schema at PowerBI

Data Visualization

Objective 1: Identify the most disciplined and undisciplined employees and divisions.



Identify the most disciplined and undisciplined employees and divisions;

1. Female employees had the highest total Tardy at 1184,
2. Assistant Pharmacist had the highest Punctual (3134) and Tardy (450).
3. The most undisciplined employee and department are User ID 75218 (116) and Pharmacy (665), respectively.
4. The most disciplined employees are User IDs 129675, 75834, 84490 & 93606 with only 1 recorded Tardy. PBF is the most disciplined department with 79 Tardy.

Figure 8. Power BI Dashboard for Objective no. 1

Findings

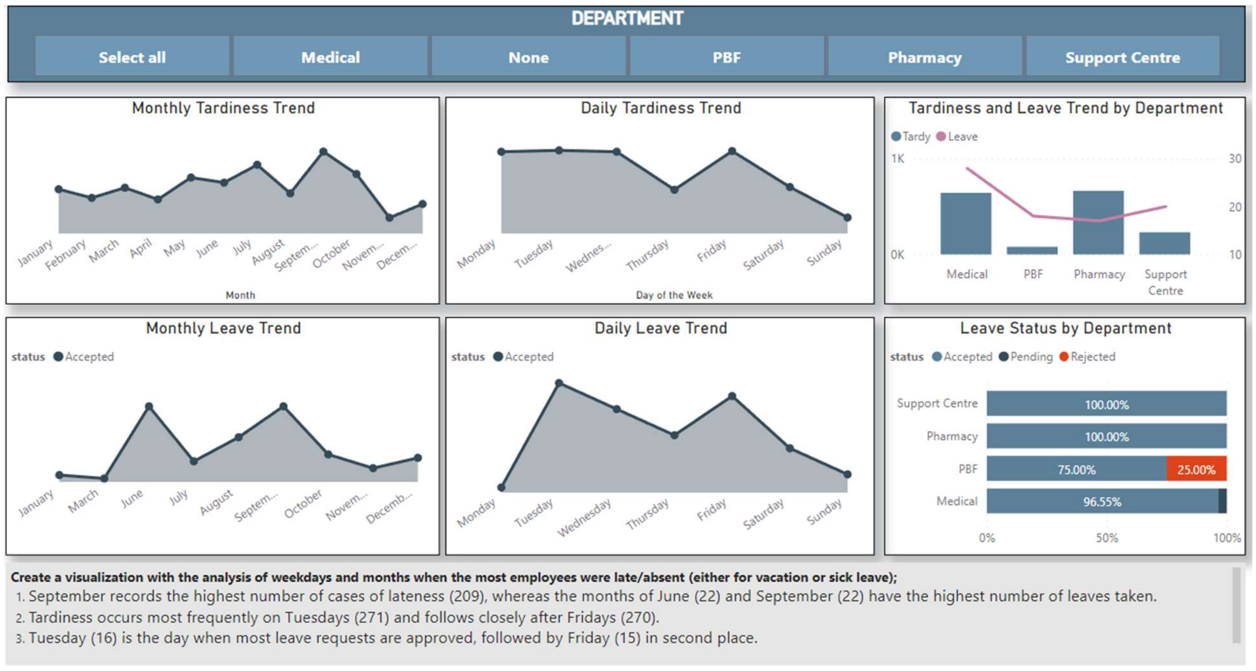
In terms of tardiness as shown in Figure 8, female employees have recorded the highest number of instances, with a total of 1184 reported cases.

The Assistant Pharmacist position stands out as the role with the highest number of punctual records, with 3134 recorded instances, as well as the highest number of tardiness records, with 450 reported cases.

On the other hand, the least disciplined employee is User ID 75218, with 116 recorded instances of tardiness, and the Pharmacy department stands out as the department with the most frequent tardiness records, with a total of 665 cases.

In contrast, User IDs 129675, 75834, 84490, and 93606 are the most disciplined employees, with only one recorded instance of tardiness each. The most disciplined department is PBF, with a total of 79 reported cases of tardiness.

Objective 2: Create a visualization with the analysis of weekdays and months when the most employees were late/absent (either for vacation or sick leave)


DEPARTMENT
Select all
Medical
None
PBF
Pharmacy
Support Centre
Monthly Tardiness Trend
January February March April May June July August Septem... October Novem... Decem...
Month
Daily Tardiness Trend
Monday Tuesday Wednes... Thursday Friday Saturday Sunday
Day of the Week
Tardiness and Leave Trend by Department
● Tardy ● Leave
1K
0K
30
20
10
Medical PBF Pharmacy Support Centre
Monthly Leave Trend
status ● Accepted
January March June July August Septem... October Novem... Decemb...
Daily Leave Trend
status ● Accepted
Monday Tuesday Wednesday Thursday Friday Saturday Sunday
Leave Status by Department
status ● Accepted ● Pending ● Rejected
Support Centre 100.00%
Pharmacy 100.00%
PBF 75.00% 25.00%
Medical 96.55%
0% 50% 100%

Create a visualization with the analysis of weekdays and months when the most employees were late/absent (either for vacation or sick leave);

1. September records the highest number of cases of lateness (209), whereas the months of June (22) and September (22) have the highest number of leaves taken.
2. Tardiness occurs most frequently on Tuesdays (271) and follows closely after Fridays (270).
3. Tuesday (16) is the day when most leave requests are approved, followed by Friday (15) in second place.

Figure 9. Power BI Dashboard for Objective no. 2

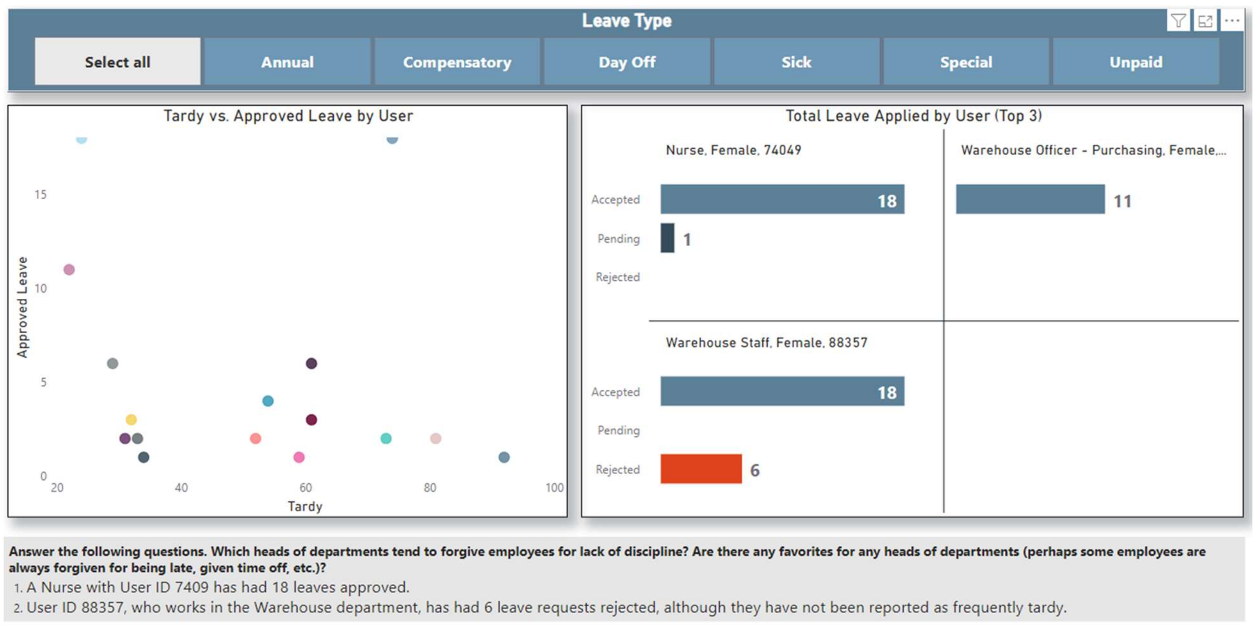
Findings

Regarding employee attendance as shown in FIgure 8, September is the month with the highest number of recorded cases of tardiness, with a total of 209 instances. Additionally, the months of June and September both have the highest number of leaves taken, with 22 recorded instances each.

When it comes to daily tardiness patterns, Tuesdays stand out as the day with the highest frequency of tardiness, with 271 recorded instances, followed closely by Fridays with 270 instances.

On the other hand, Tuesday is the day when most leave requests are approved, with a total of 16 recorded instances, followed by Friday with 15 instances, coming in second place in terms of leave approval (refer to Figure 9).

Objective 3: Answer the following questions. Which heads of departments tend to forgive employees for lack of discipline? Are there any favorites for any heads of departments (perhaps some employees are always forgiven for being late, given time off, etc.)?



Leave Type
Select all
Annual
Compensatory
Day Off
Sick
Special
Unpaid
Tardy vs. Approved Leave by User
Approved Leave
Tardy
Total Leave Applied by User (Top 3)
Nurse, Female, 74049
Warehouse Officer - Purchasing, Female,...
Warehouse Staff, Female, 88357
Accepted
Pending
Rejected
18
1
11
18
6

Answer the following questions. Which heads of departments tend to forgive employees for lack of discipline? Are there any favorites for any heads of departments (perhaps some employees are always forgiven for being late, given time off, etc.)?

1. A Nurse with User ID 7409 has had 18 leaves approved.
2. User ID 88357, who works in the Warehouse department, has had 6 leave requests rejected, although they have not been reported as frequently tardy.

Figure 10. Power BI Dashboard for Objective no. 3

Findings

Both charts in Figure 10 indicate that some departments may face favoritism. Because the tardiest employee, User ID 74049, a Nurse, has 18 approved leaves. Employee 88357, a Warehouse Staff member, is on the left end of the spectrum with 6 Rejected leaves but is not as tardy.

**Note: We filtered Top 3 for report purposes, and Top 10 for PowerBI Report because it is interactive*.

Recommendations

Based on the findings of the report, the following recommendations can be made:

1. Disciplinary measures should be implemented for employees who frequently arrive late or leave early, especially for those in the Assistant Pharmacy position.
2. We can Implement a performance improvement plan for the Assistant Pharmacist position, which has the highest number of tardiness records despite also having the highest number of punctual records. This may include targeted training or coaching on time management and prioritization.
3. The HR department should monitor employee leave trends, particularly in June, when the Medical Department has a higher rate of approved leave, and also in September, when both the Medical and Support Centre have a higher rate of approved leave. These two

months mark the beginning and the end of the dry (summer) season for Indonesia, and some employees may be taking leaves for summer outings with family and friends before the rainy season begins in October. On this note, if productivity will be significantly affected, it is possible to consider seasonal employees to fill in for the expected absences as leave requests will have been filed.

4. It may also be noted that tardiness peaks on Fridays as well, working undertime or leaving early for the weekend may be the reason here. A strong reminder with possible disciplinary measures to all employees would be a good first step to curb this practice.

5. Heads of departments should avoid showing favoritism towards certain employees and ensure that all employees are treated equally. The HR department should also monitor the approval of leave requests to ensure that there is no bias in the approval process.

6. Further investigation can be done to determine the reasons behind the trends identified in the report, such as the high number of approved leaves in the Nurse and Warehouse department and gender differences in tardiness. This can aid in addressing any underlying issues affecting employee productivity and discipline.