Data Analysis on Human Resource Problems for a Medical Company (Project Clockster)

Group 17

Lacsamana, Francis Marvin

Manaois, Marina

Marable, Alicia

Rebuta, Michael

Reyes, Fritz Gerald

Taller, Rommel

Submitted 31 March 2023

Contents

Problem

Problem Statement & Data Analysis Goals

The medical company's HR department wants to determine insights on employee attendance. In line with this, the CEO formulated following questions:

1. Identify the most disciplined and undisciplined employees and divisions.
2. Create a visualization with the analysis of weekdays and months when the most employees were late/absent (either for vacation or sick leave).
3. 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.?)

The data provided by HR department includes 10-15 parameters per day per year (arrival/departure time, vacations, sick days, time off, etc.) on 1000+ employees of the company.

Company Profile

Clockster was founded in 2017 by the joint efforts of four tech enthusiasts to solve employee salary overpay problems based on paper timesheets for Americana Group after securing $100K in angel investments from a Kazakh businessman Abdrakhman Amreyev.

After the initial B2B sales in 2018 it was decided to become a fully realized SaaS and shift focus on convenience of having mobile and web applications. In the subsequent year, Clockster raised $240K in seed investments of from US-KZ VC fund ABC-I2BF and Singapore-based Kazakh businessman Olzhas Zhiyenkulov (Paladigm Capital, CEO).

Clockster continued serving its clients during the COVID-19 pandemic and joined the global battle by introducing mask and temperature recognition to the access control terminals.

By staying right on course Clockster ends 2020 with a $750K funding round led by Singaporean Quest Ventures and HR&ED-tech accelerator program to support expansion to SEA.

Clockster's rapid growth is being noticed and recognized not only by SEA VCs but is acknowledged by one of the world's leading venture companies, 500 Global. After joining its acceleration program (SF batch 29), Clockster team receives another round of investments for an even more aggressive presence establishment in Indonesia.

Through partnership and networking, Clockster unveils a great market potential in Uzbekistan. In almost one year client database increases by 100+ clients.

After moving its HQ to Jakarta, Indonesia Clockster quickly becomes a hit with F&B and retail companies like Yellowcarwash, Etika Beverages, Kick Avenue, Nama Beauty and many more.

Methodology

Data Sources, Tools, and Procedure

The recommendations and data analysis included in this report were derived from Clockster's available HR data from Oct 2021 to Oct 2022. The available data points are described in the Glossary of Terms.

All the analysts in the team inspected the dataset individually to look for issues in the dataset prior to analysis.

Pre-Cleaning (MS Excel)

Upon inspection of dataset:

#	CSV File	Notes
1	attendance.csv	
2	leave_requests.csv	needs pre-cleaning
3	payroll.csv	
4	schedules.csv	needs pre-cleaning
5	users.csv	

We noticed that some of these cannot be directly imported to PostgreSQL due to inconsistencies in data formatting, specifically the array within some of the columns.

We used MS Excel to perform pre-cleaning of CSV files prior import to PostgreSQL.

Notice that the following changes were done to fix the formatting to allow import in PostgreSQL:

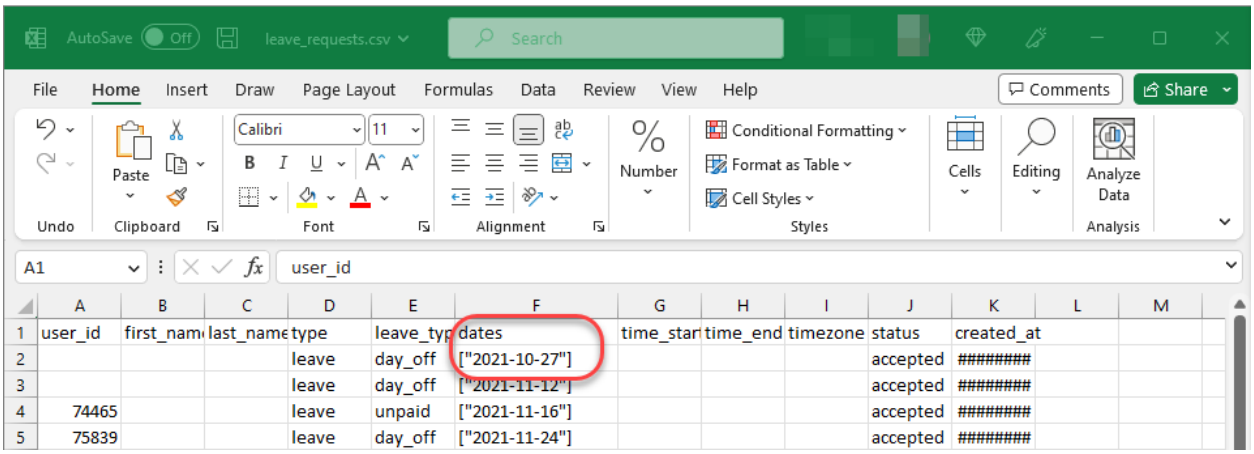
CSV File	Column	From	To
leave_requests.csv	dates	• Square Bracket or [ ] • Double Quote or “	• Curly Brace or { } • Single Quote or ‘
schedules.csv	dates	• Square Bracket or [ ] • Double Quote or “	• Curly Brace or { } • Single Quote or ‘
	user_id		• Added a Single Quote as a delimiter

Other CSV files do not require pre-cleaning and can be imported directly in PostgreSQL.

Below are sample screenshots.

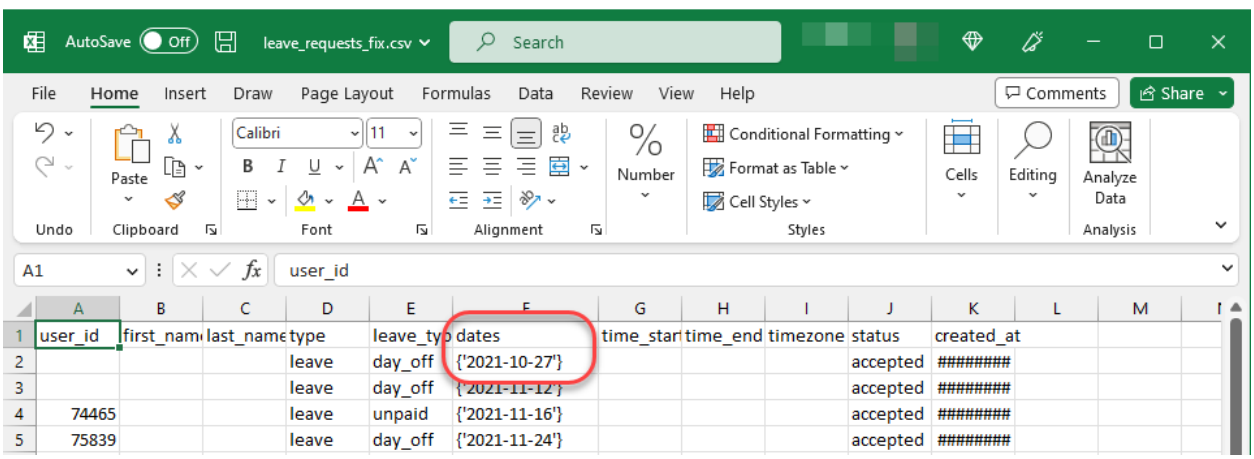
leave_requests.csv

Original:



	A	B	C	D	E	F	G	H	I	J	K
1	user_id	first_nam	last_name	type	leave_typ	dates	time_star	time_end	timezone	status	created_at
2				leave	day_off	["2021-10-27"]				accepted	########
3				leave	day_off	["2021-11-12"]				accepted	########
4	74465			leave	unpaid	["2021-11-16"]				accepted	########
5	75839			leave	day_off	["2021-11-24"]				accepted	########

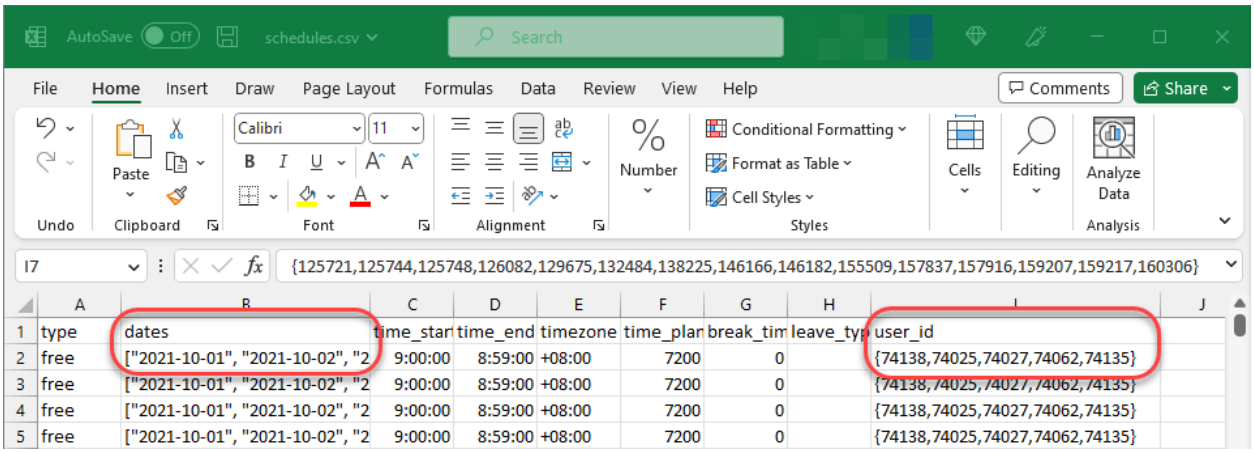
Fixed:



	A	B	C	D	E	F	G	H	I	J	K
1	user_id	first_nam	last_name	type	leave_typ	dates	time_star	time_end	timezone	status	created_at
2				leave	day_off	{'2021-10-27'}				accepted	########
3				leave	day_off	{'2021-11-12'}				accepted	########
4	74465			leave	unpaid	{'2021-11-16'}				accepted	########
5	75839			leave	day_off	{'2021-11-24'}				accepted	########

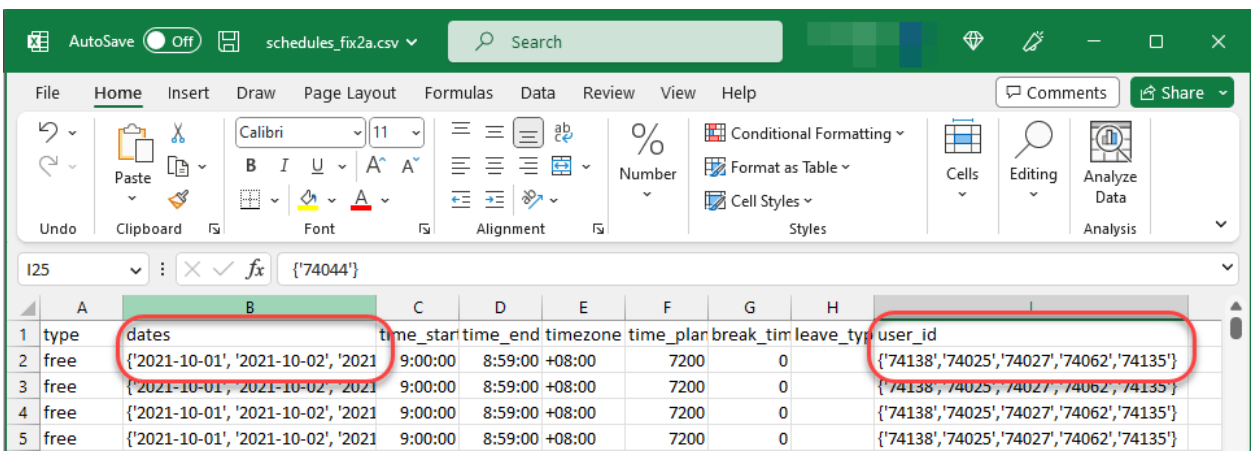
schedules.csv

Original:



	A	B	C	D	E	F	G	H	I
1	type	dates	time_star	time_end	timezone	time_plan	break_tim	leave_typ	user_id
2	free	["2021-10-01", "2021-10-02", "2	9:00:00	8:59:00	+08:00	7200	0		{74138,74025,74027,74062,74135}
3	free	["2021-10-01", "2021-10-02", "2	9:00:00	8:59:00	+08:00	7200	0		{74138,74025,74027,74062,74135}
4	free	["2021-10-01", "2021-10-02", "2	9:00:00	8:59:00	+08:00	7200	0		{74138,74025,74027,74062,74135}
5	free	["2021-10-01", "2021-10-02", "2	9:00:00	8:59:00	+08:00	7200	0		{74138,74025,74027,74062,74135}

Fixed:



	A	B	C	D	E	F	G	H	I
1	type	dates	time_star	time_end	timezone	time_plan	break_tim	leave_typ	user_id
2	free	{'2021-10-01', '2021-10-02', '2021	9:00:00	8:59:00	+08:00	7200	0		{'74138','74025','74027','74062','74135'}
3	free	{'2021-10-01', '2021-10-02', '2021	9:00:00	8:59:00	+08:00	7200	0		{'74138','74025','74027','74062','74135'}
4	free	{'2021-10-01', '2021-10-02', '2021	9:00:00	8:59:00	+08:00	7200	0		{'74138','74025','74027','74062','74135'}
5	free	{'2021-10-01', '2021-10-02', '2021	9:00:00	8:59:00	+08:00	7200	0		{'74138','74025','74027','74062','74135'}

The following Excel functions were used:

- Find & Replace
- Text to Column & Concatenate

Data Cleaning (PostgreSQL)

From the attachment files, you may inspect the SQL Scripts to check the following workflow:

- Create Schema
- Create Tables
- Create Copy Tables (with name prefix “c”)
- Data Cleanup using UPDATE statement (on “c” prefixed tables)
- UNNEST of the following tables (cschedules & cleaverequest) – due to multiple values in an array on some of its columns, UNNESTed tables are denoted with name prefix “u_”, a prefix of “u2_” means we used UNNEST 2 times, due to 2 columns having array values
- Using DISTINCT to show interesting categorical data points for potential insights
- Using DISTINCT to remove duplicates on the datasets (on cattendance & cschedules), denoted by name prefix “dedup_”
- Below are the tables where we exported the data for import in Power BI:
    - cusers
    - cpayroll
    - u_cleaverequest
    - dedup_cattendance
    - dedup_u2_cschedules

Below are sample screenshots of an array:

SELECT * FROM gp2.leaverequest

	user_id	first_name	last_name	type	leave_type	dates	time_start	time_end	timezone	status	created_at
1	[NULL]	[NULL]	[NULL]	leave	day_off	{2021-10-27}	[NULL]	[NULL]	[NULL]	accepted	2021-10-25 12:38:00.000
2	[NULL]	[NULL]	[NULL]	leave	day_off	{2021-11-12}	[NULL]	[NULL]	[NULL]	accepted	2021-11-07 15:20:00.000
3	74465	[NULL]	[NULL]	leave	unpaid	{2021-11-16}	[NULL]	[NULL]	[NULL]	accepted	2021-11-15 13:45:00.000
4	75839	[NULL]	[NULL]	leave	day_off	{2021-11-24}	[NULL]	[NULL]	[NULL]	accepted	2021-11-16 02:18:00.000
5	74042	[NULL]	[NULL]	leave	day_off	{2021-11-28}	[NULL]	[NULL]	[NULL]	accepted	2021-11-27 07:06:00.000
6	74054	[NULL]	[NULL]	leave	sick	{2021-12-23}	[NULL]	[NULL]	[NULL]	accepted	2021-12-23 04:04:00.000
7	74049	[NULL]	[NULL]	leave	sick	{2021-12-09,2021-12-10,2021-12-11, ... [6]}	[NULL]	[NULL]	[NULL]	accepted	2021-12-27 13:50:00.000
7.1						2021-12-09					
7.2						2021-12-10					
7.3						2021-12-11					
7.4						2021-12-12					
7.5						2021-12-13					
7.6						2021-12-14					

SELECT * FROM gp2.schedules

	type	dates	time_start	time_end	timezone	time_planned	break_time	leave_type	user_id
1	free	{2021-10-01,2021-10-02,2021-10-04, ... [392]}	09:00:00	08:59:00	+08:00	7,200	0	[NULL]	{'74138','74025','74027', ... [5]}
2	free	{2021-10-01,2021-10-02,2021-10-04, ... [392]}	09:00:00	08:59:00	+08:00	7,200	0	[NULL]	{'74138','74025','74027', ... [5]}
3	free	{2021-10-01,2021-10-02,2021-10-04, ... [392]}	09:00:00	08:59:00	+08:00	7,200	0	[NULL]	{'74138','74025','74027', ... [5]}
4	free	{2021-10-01,2021-10-02,2021-10-04, ... [392]}	09:00:00	08:59:00	+08:00	7,200	0	[NULL]	{'74138','74025','74027', ... [5]}
5	work	{2021-11-15}	15:30:00	21:30:00	+08:00	21,600	0	[NULL]	{'75834'}
6	work	{2021-11-07}	15:30:00	21:30:00	+08:00	21,600	0	[NULL]	{'74053'}
7	work	{2021-11-13,2021-11-14,2021-11-21, ... [5]}	08:30:00	14:30:00	+08:00	21,600	0	[NULL]	{'74053'}
8	work	{2021-11-15,2021-11-18,2021-11-22}	15:30:00	21:30:00	+08:00	21,600	0	[NULL]	{'75848'}
9	work	{2021-11-04,2021-11-11,2021-11-12, ... [6]}	15:30:00	21:30:00	+08:00	21,600	0	[NULL]	{'75839'}
10	work	{2021-11-10,2021-11-11,2021-11-19, ... [4]}	15:30:00	21:30:00	+08:00	21,600	0	[NULL]	{'75955'}
11	work	{2021-11-24}	08:30:00	14:30:00	+08:00	21,600	0	[NULL]	{'75839'}
12	free	{2021-10-01,2021-10-02,2021-10-04, ... [392]}	09:00:00	08:59:00	+08:00	7,200	0	[NULL]	{'74138','74025','74027', ... [5]}
12.1		2021-10-01							'74138'
12.2		2021-10-02							'74025'
12.3		2021-10-04							'74027'
12.4		2021-10-05							'74062'
12.5		2021-10-06							'74135'
12.6		2021-10-07							
12.7		2021-10-08							
12.8		2021-10-09							
12.9		2021-10-11							
12.10		2021-10-12							
12.11		2021-10-13							
12.12		2021-10-14							
12.13		2021-10-15							

Below is the screenshot of tables in PostgreSQL: *highlighted are the tables to be exported and to be imported in Power BI*

- refocus_exercises - *localhost:5432*
  - Databases
    - **refocus_exercises**
      - Schemas
        - gp1
        - gp2
          - Tables
            - attendance
            - cattendance
            - cleaverequest
            - cpayroll
            - cschedules
            - cusers
            - dedup_cattendance
            - dedup_u2_cschedules
            - leaverequest
            - payroll
            - schedules
            - u2_cschedules
            - u_cleaverequest
            - u_cschedules
            - users

The following issues in the data were found and has been amended.

1. Spellings and misspellings for categorical data points

Table Name	**Column Name**	**Notes**
users	position	Indonesian terms were replaced with English term translation
attendance	location	Upper case "OSADHA BELEGA" replaced with "Osadha Belega"

2. Null or empty values

Table Name	**Column Name**	**Notes**
users	gender	Blanks were replaced with "not specified"
users	department	Blanks were replaced with "not specified"
users	employment	Blanks were replaced with "full_time" (assumption)
schedules	leave_type	Blanks were replaced with "not specified"
payroll	currency	Blanks were replaced with "IDR"
attendance	location	Blanks were replaced with "not specified"

3. Duplicated entries

Table Name	**Notes**
attendance	DISTINCT records were selected to clean the duplicates
schedules	DISTINCT records were selected to clean the duplicates

Analysis (Power BI)

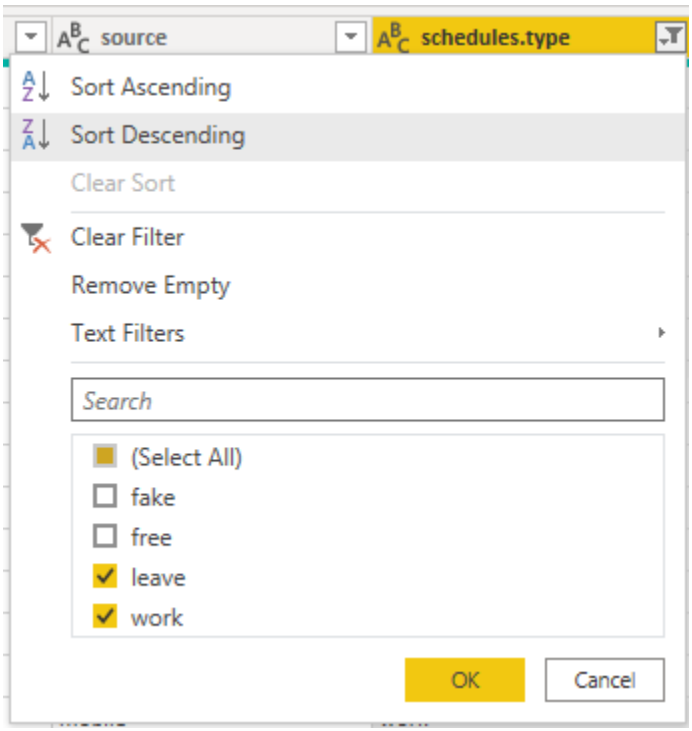
Below is the workflow on Power BI:

- Import Data from CSV exported from PostgreSQL, review data types
- Merge attendance & schedule tables as **attendance_schedule** table, use inner join
- Calculate DAX metrics (Add Column)
- Create **calendar** table using MIN & MAX on attendance_schedule merged table

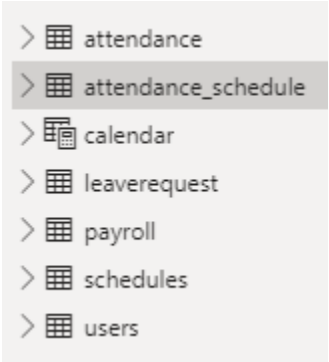
Merge Query in Power Query (**attendance** & **schedule** -> **attendance_schedule**)

```
let
    Source = Table.NestedJoin(attendance, {"user_id", "date"}, schedules, {"user_id", "dates"},
    "schedules", JoinKind.Inner),
    #"Expanded schedules" = Table.ExpandTableColumn(Source, "schedules", {"type",
    "time_start", "time_end", "timezone", "time_planned", "break_time", "leave_type"},
    {"schedules.type", "schedules.time_start", "schedules.time_end", "schedules.timezone",
    "schedules.time_planned", "schedules.break_time", "schedules.leave_type"}),
    #"Filtered Rows" = Table.SelectRows(#"Expanded schedules", each ([schedules.type] = "leave"
    or [schedules.type] = "work")),
    #"Merged Queries" = Table.NestedJoin(#"Filtered Rows", {"user_id"}, users, {"user_id"},
    "users", JoinKind.Inner),
    #"Expanded users" = Table.ExpandTableColumn(#"Merged Queries", "users", {"department"},
    {"users.department"})
in
    #"Expanded users"
```

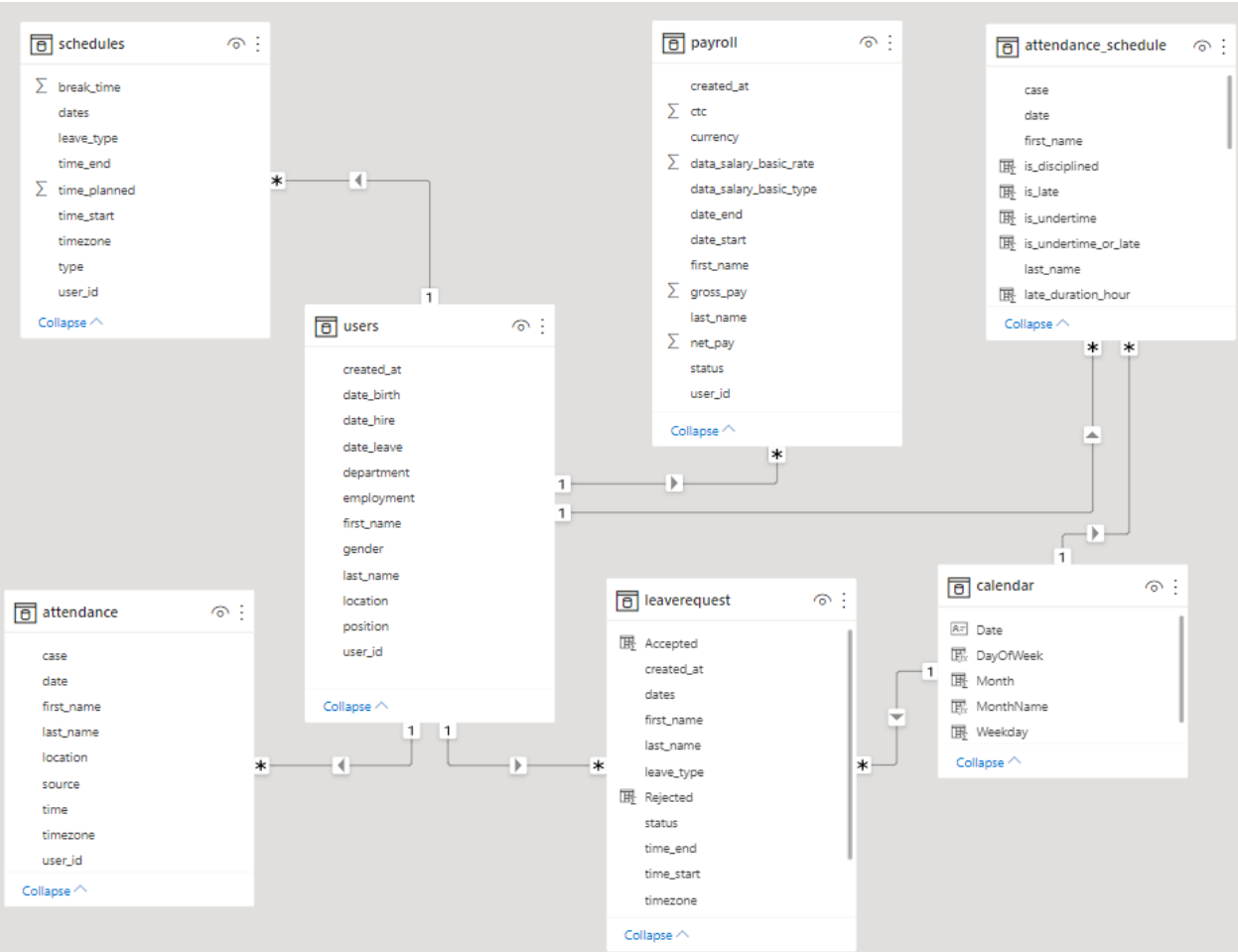
Note that we filtered out **fake** & **free** in **type** column of schedules (only selected **work** and **leave**).



Tables



Data Model


schedules
break_time
dates
leave_type
time_end
time_planned
time_start
timezone
type
user_id
Collapse
payroll
created_at
ctc
currency
data_salary_basic_rate
data_salary_basic_type
date_end
date_start
first_name
gross_pay
last_name
net_pay
status
user_id
Collapse
attendance_schedule
case
date
first_name
is_disciplined
is_late
is_undertime
is_undertime_or_late
last_name
late_duration_hour
Collapse
users
created_at
date_birth
date_hire
date_leave
department
employment
first_name
gender
last_name
location
position
user_id
Collapse
attendance
case
date
first_name
last_name
location
source
time
timezone
user_id
Collapse
leaverequest
Accepted
created_at
dates
first_name
last_name
leave_type
Rejected
status
time_end
time_start
timezone
Collapse
calendar
Date
DayOfWeek
Month
MonthName
Weekday
Collapse

DAX Codes

Table	New Column	Notes
attendance_schedule	undertime_duration_minute = IF([case] = "IN", DATEDIFF([schedules.time_start], [time], MINUTE), IF([case] = "OUT", DATEDIFF([time], [schedules.time_end], MINUTE)))	Time difference calculation whether late or undertime, depending if case is IN or OUT
	late_duration_hour = IF([case] = "IN", DATEDIFF([schedules.time_start], [time], HOUR), 0)	If case is IN, calculate the time difference in HOUR
	late_gt_2hr = IF([case] = "IN", IF([late_duration_hour] > 2, "TRUE", "FALSE"), "N/A")	This is used to filter the visualization, late for more than 2 hours is considered an outlier
	is_undertime = IF([case] = "OUT", IF([undertime_duration_minute] > 0, 1, 0), 0)	Put a 1 in this column if undertime duration is greater than 0
	is_late = IF([case] = "IN", IF([undertime_duration_minute] > 10, 1, 0), 0)	Put a 1 in this column if late duration is greater than 10
	is_undertime_or_late = [is_undertime] + [is_late]	Adds the value of the is_undertime and is_late column
	is_disciplined = IF([is_undertime_or_late] = 0, 1, 0)	If is_undertime_or_late is set to 0, sets this column to 1, else 0
	undiscipline_magnitude = IF(AND(attendance_schedule[is_undertime] = 1, attendance_schedule[is_late] = 1),2,	Case by case basis, sets the magnitude of

	IF(AND(attendance_schedule[is_undertime] = 1,
	attendance_schedule[is_late] = 0),1,
	IF(AND(attendance_schedule[is_undertime] = 0,
	attendance_schedule[is_late] = 1),1,
	IF(AND(attendance_schedule[is_undertime] = 0,
	attendance_schedule[is_late] = 0),0
	))))
		being undisciplined

	year_date = YEAR([date])	Gets the YEAR value of date
	month_date = MONTH([date])	Gets the MONTH value of date
	percent_tardiness = [is_undertime_or_late] / 365	Percent Tardiness Counter in a Year
calendar	calendar = CALENDAR(MIN('attendance_schedule'[date]),MAX('attendance_schedule'[date]))	Create calendar table using MIN & MAX on attendance_schedule merged table
	Weekday = WEEKDAY([Date],2)	Calculate the Weekday
	WeekNum = WEEKNUM([Date])	Calculate the Week Number

	DayOfWeek = IF([Weekday] = 1, "1 (Mon)",
	IF([Weekday] = 2, "2 (Tue)",
	IF([Weekday] = 3, "3 (Wed)",
	IF([Weekday] = 4, "4 (Thu)",
	IF([Weekday] = 5, "5 (Fri)",
	IF([Weekday] = 6, "6 (Sat)",
	IF([Weekday] = 7, "7 (Sun)")))))))
		Calculate the Day of Week

	Month = MONTH([Date])	Calculate the Month

	MonthName = IF([Month] = 1, "01 (Jan)",
	IF([Month] = 2, "02 (Feb)",
	IF([Month] = 3, "03 (Mar)",
	IF([Month] = 4, "04 (Apr)",
	IF([Month] = 5, "05 (May)",
	IF([Month] = 6, "06 (Jun)",
	IF([Month] = 7, "07 (Jul)",
	IF([Month] = 8, "08 (Aug)",
		Calculate the Month (Month Number & Name)

	IF([Month] = 9, "09 (Sep)",
	IF([Month] = 10, "10 (Oct)",
	IF([Month] = 11, "11 (Nov)",
	IF([Month] = 12, "12 (Dec)"))))))))))))	

leaverequest	year_of_leave_request = YEAR([dates])	Calculate the Year
	Rejected = IF(leaverequest[status] = "accepted",0,if(leaverequest[status] = "pending",0,1))	Counter for rejected leave request
	Accepted = IF(leaverequest[status] = "rejected",0,if(leaverequest[status] = "pending",0,1))	Counter for accepted leave request

Then, analysis and visualizations are created as well in Power BI.

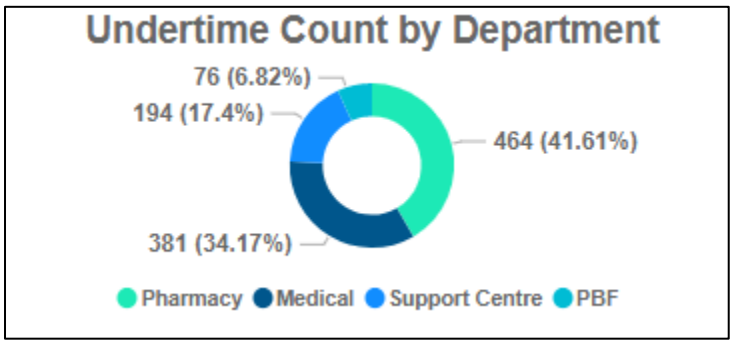
Findings

Disciplined and Undisciplined Employees and Divisions

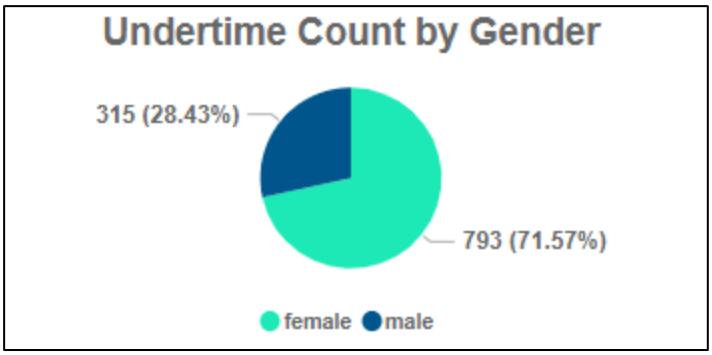
Note:

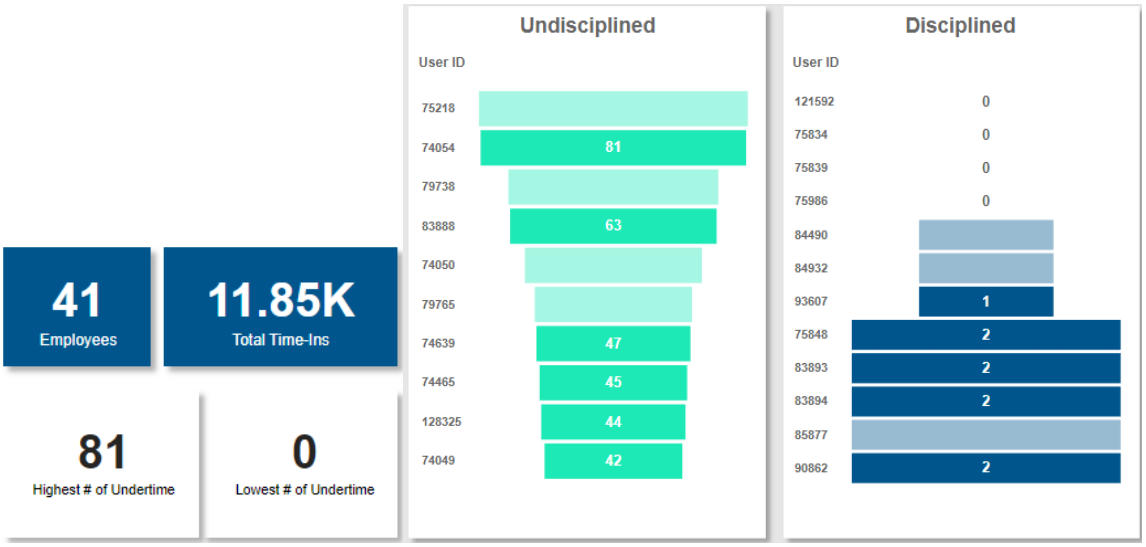
The term “undertime” in this report means both late and undertime.

- Pharmacy has the most cases of undertime shifts, almost 6x the PBF which has the least cases.

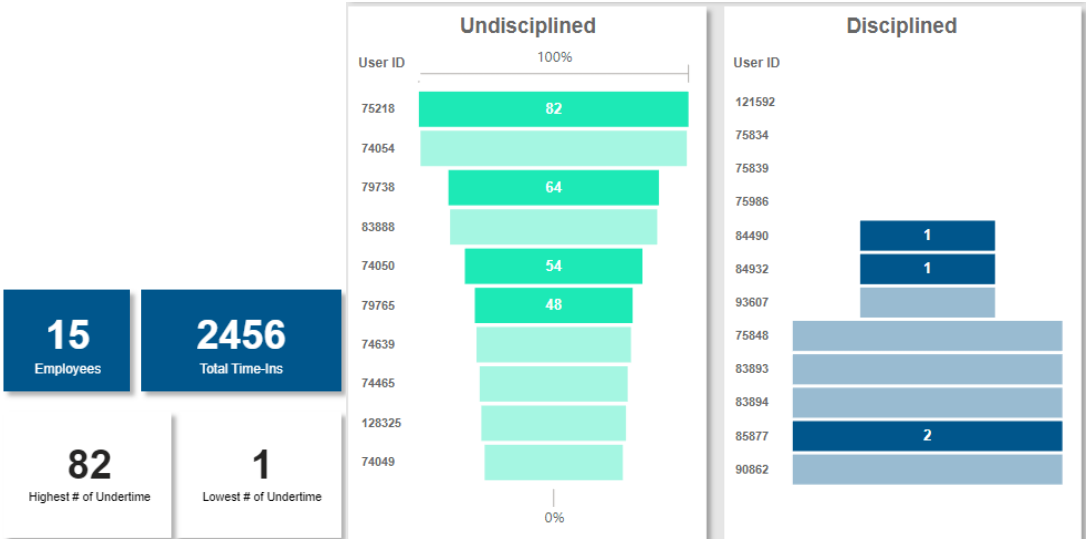


- Majority (71.57%) of the undertime cases are from female employees.



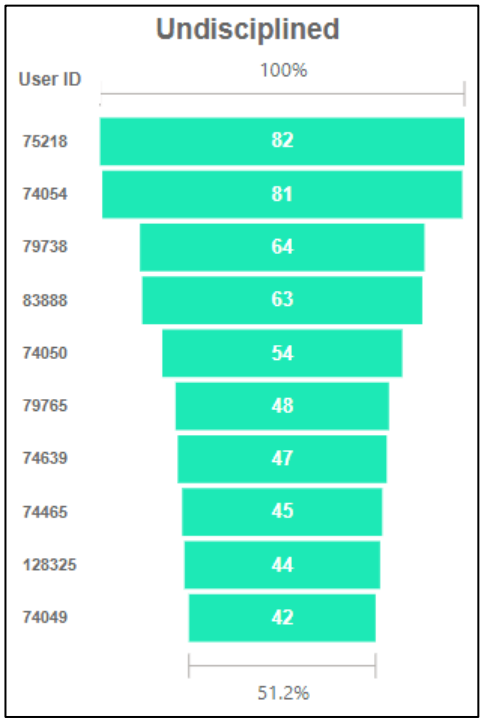


Female data – Comprising most (41 employees) of the sample size. Comprising of 4 top disciplined employees are all females, having 0 undertime/late. And comprising 9 out of the 12 most disciplined employes.

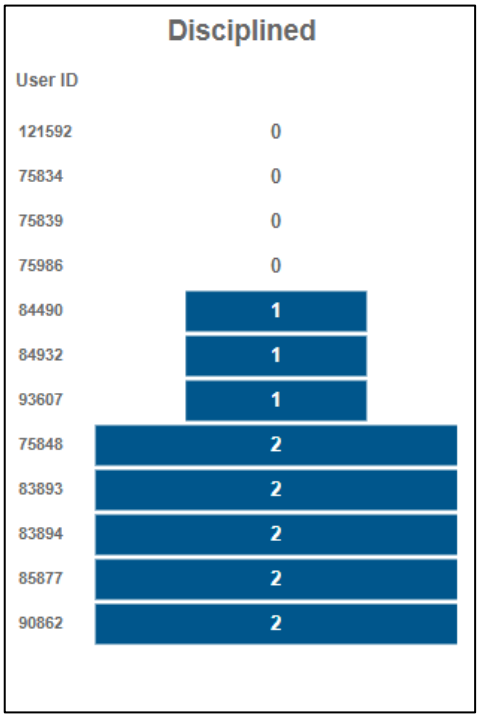


Male Data - Comprising less than half (15 employees) of the sample size. And comprising 3 out of the 12 most disciplined employes.

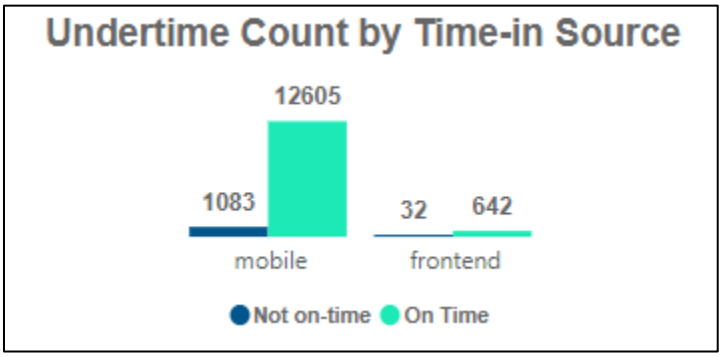
- The most undisciplined employee, i.e. the employee with the most undertime counts, is User 75218, an IT Supervisor from the Support Centre.



- The most disciplined employees are Users 121592, 75834, 75839, and 75986 with 0 undertime counts.

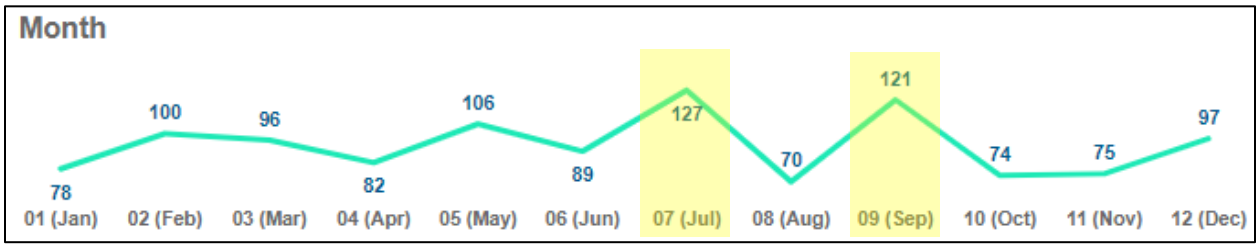


- The most used time-in source is Mobile where the undertime count is 12605.

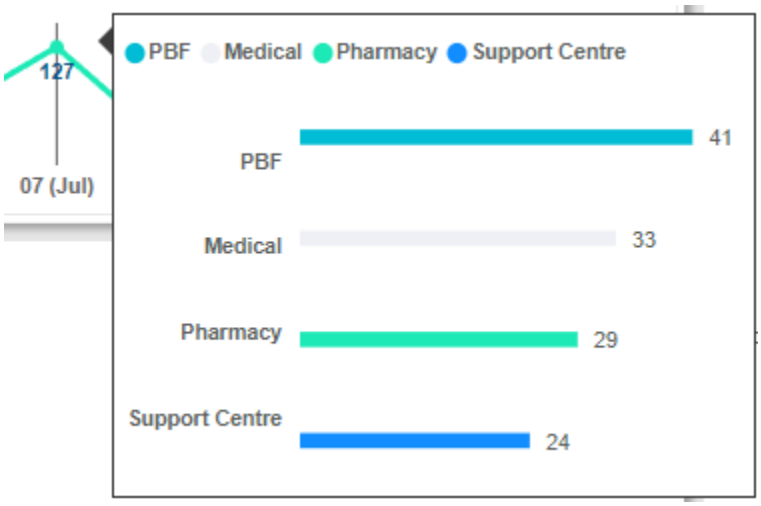


Peaks in Late and Absent Cases

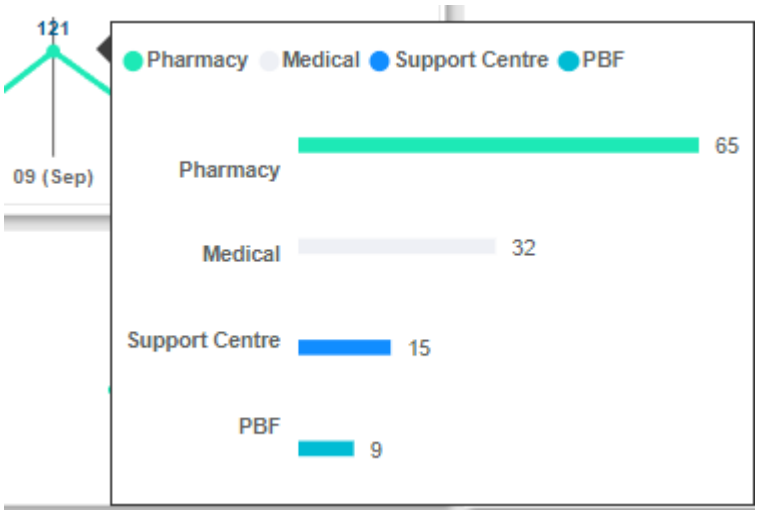
- In the year, July and September have the most undertime cases.



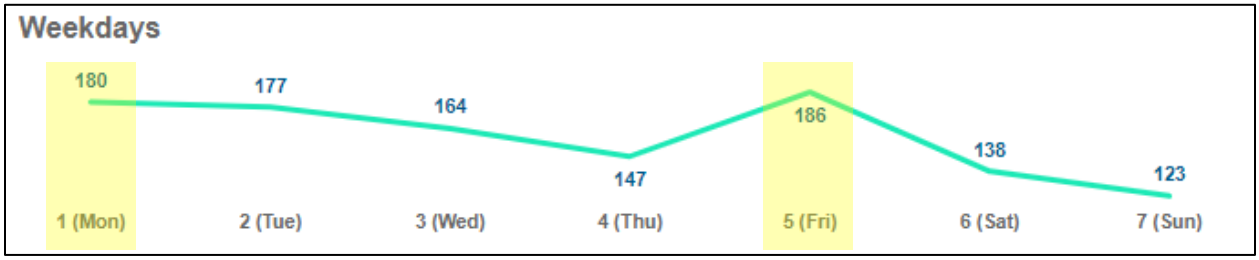
On the Month of July, PBF has the highest Undertime Count (41) followed by Medical.



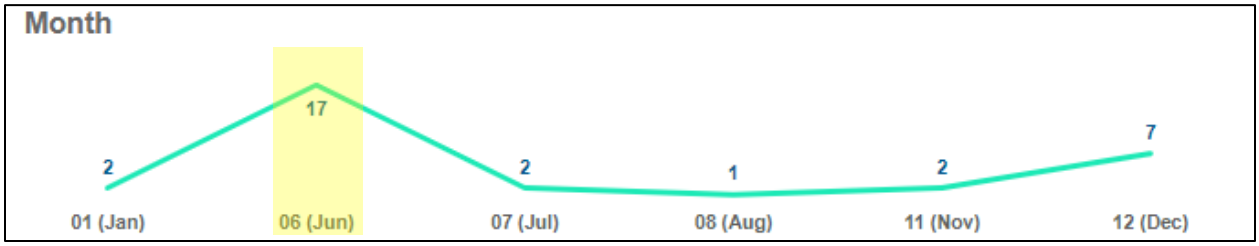
On the Month of September, Pharmacy has the highest Undertime Count (65).

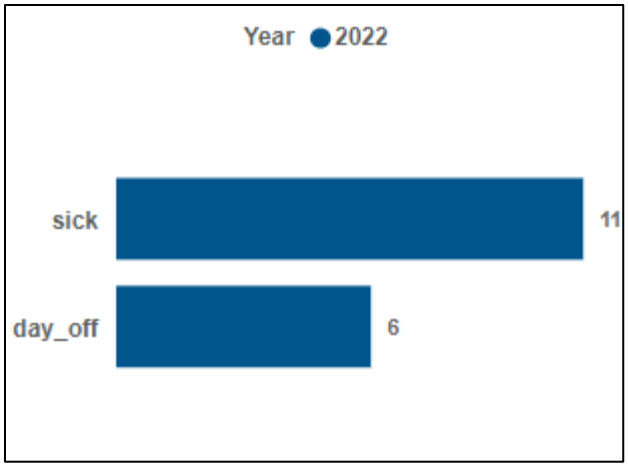


- Undertime cases peak on Mondays and Fridays



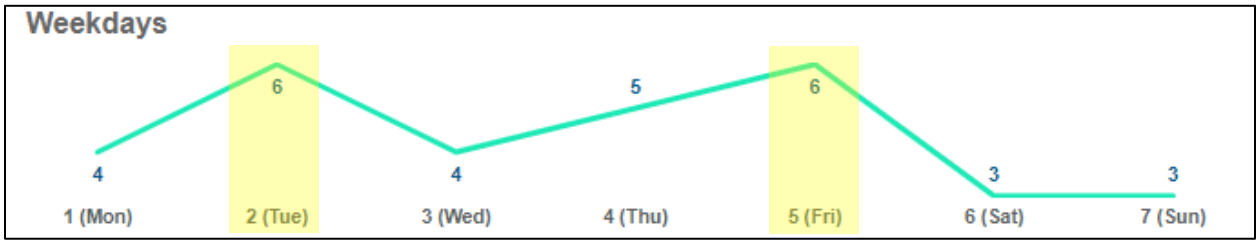
- In the year, June has the most leave requests, 2/3 of which are sick leaves.



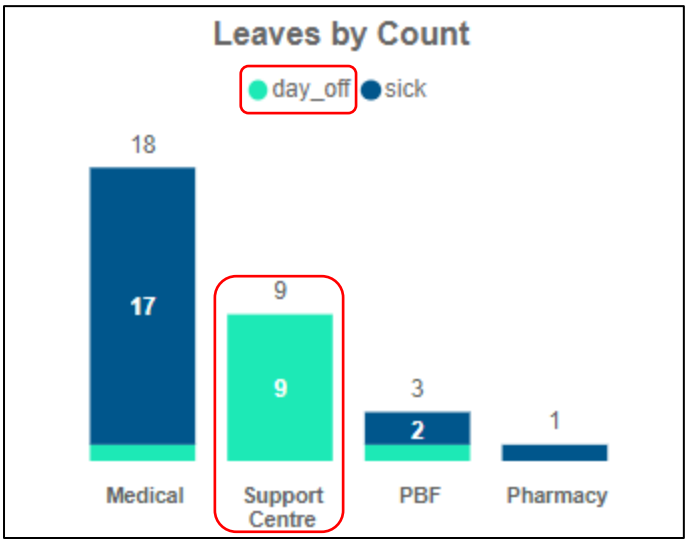


On the Month of June, sick leave is almost double the amount of day off requests.

- Leave requests peak on Tuesdays and Fridays.



- Most day off requests come from the Support Centre.



Suspected Favoritism

- Most accepted leave requests are from the Medical department, while the department with the most rejected leave requests are from the PBF department.

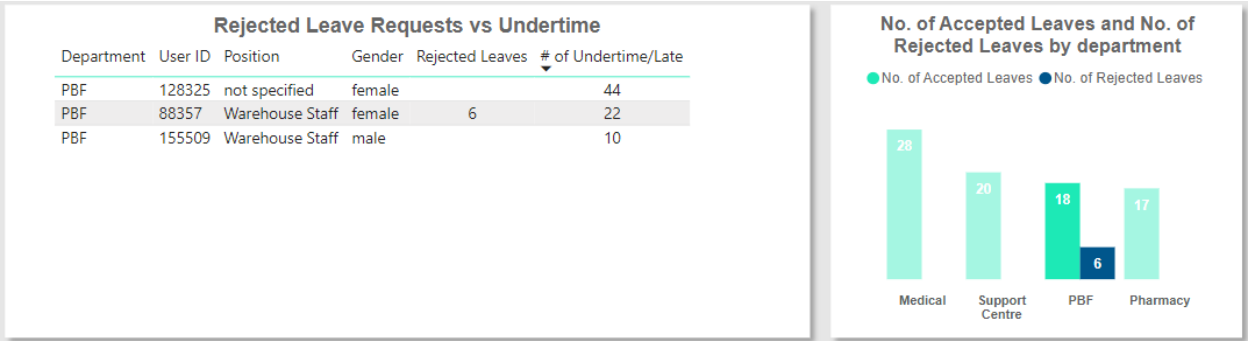
- Users 75218 and 74054 have the highest tardiness %, being late for 80 days in a year
  - Despite their tardiness record, User 74054 had all their six leave requests accepted.

Department	User ID	Position	Gender	Accepted Leaves	# of Undertime/Late
Support Centre	75218	IT Supervisor	male		82
Pharmacy	74054	Pharmacist Assistant	female	6	81

- User 74049 had 18 leave requests accepted and was tardy 42 days in the year.

Department	User ID	Position	Gender	Accepted Leaves	# of Undertime/Late
Medical	74049	Nurse	female	18	42

- Drilling down on PBF department, the table below shows only 1 employee was having leave requests rejected.



Rejected Leave Requests vs Undertime

Department	User ID	Position	Gender	Rejected Leaves	# of Undertime/Late
PBF	128325	not specified	female		44
PBF	88357	Warehouse Staff	female	6	22
PBF	155509	Warehouse Staff	male		10

Recommendations

1. **Establish clearer tardiness policies for full-time employees.**
    - Attendance policy is needed in an organization when frequent employee absences and tardiness are causing disturbance in the business.
    - To deal with employee issues, time and attendance policy is needed.
2. **Create a sanction policy for undisciplined employees.**
    - Effective working in an organization depends on the punctuality, sincerity, and regular working of the employees, and to a certain extent on employee attendance.
    - It is necessary to lay down rules and regulations which are followed within the organization to maintain regularity.
    - Point system is one of the ways to keep control of discipline related to employee attendance and to communicate that an absence cannot be taken casually and regular absence or tardiness or absenteeism will be dealt with strict disciplinary action.
    - It is recommended that unauthorized absences should be handled with a succession of warning, which, if ignored may result in a disciplinary hearing.
    - If the employee continues unauthorized absence, it is a gross misconduct, then the employee may be dismissed for that.
3. **Reward system for consistently punctual employees.**
    - Recognizing employees for good attendance and performance can be one of the lowest cost, yet highest impact strategies for your business.
    - Find a way to call out and reward good attendance on a regular basis.
    - Incentivize rewards with good attendance record. An idea could be offering rewards they would not want to miss, like an extra day off or a chance to choose their own schedule for a week.
4. **Improve data gathering:**
    - Add data points on leave requests, e.g., reasons for granting compensatory leave.
    - For those who clocked in late, collect the reasons for their tardiness.
    - Create a separate data collection system for freelancers and full-time workers.
    - Collect data on whether they gave sanctions to undisciplined employees.
    - Research the issue by undertaking a historical study to ascertain the extent of the attendance problem and whether it is improving or worsening.
    - Further analysis can help pinpoint specific problem areas, such as geographic locations, departments or divisions experiencing higher-than-usual absence rates.

Appendix

Data Dictionary

- **attendance**
  user_id, first_name, last_name – employee description, each employee has a unique user ID
  location – worksight of employee
  date, time, timezone – timestamp of employee check in (or check out)
  case – shows if employee was starting the shift (IN), ending it (OUT) or going on break (BREAK)
  source – how the data was collected (frontend or mobile)

- **leave_requests**
  user_id, first_name, last_name – employee description, each employee has a unique user ID
  type – type of request (leave)
  leave_type – type of leave requested (sick day, special, day_off, compensatory, unpaid, annual)
  dates – dates requested for leave
  time_start, time_end, timezone – blank columns, timestamp for leave requests
  status – whether the request id accepted or rejected
  created at – timestamp of request creation by employee

- **payroll**
  user_id, first_name, last_name – employee description, each employee has a unique user ID
  date_start, date_end – period to be paid to employee
  ctc, net_pay, gross_pay – accountant values for employee's salary
  data_salary_basic_rate – employee's salary
  data_salary_basic_type – on what basis is the salary calculated (daily/monthly)
  currency – IDR (Indonesian Rupiah)
  status – whether payment was accepted or rejected
  created at – timestamp of request creation

- **schedules**
  Type – whether the employee is working or free for the specified dates
  Dates – dates
  time_start, time_end, timezone – scheduled worktime for specified dates for the employee (user ID)
  time_planned – workday (seconds)
  break_time – time for break (seconds)leave type - whether the employee is on the leave
  user ID – employee unique ID

- **users**
  Information of each employee (user_id, first_name, last_name, gender, date of birth, date of hire, date_leave, employment (full-time or part-time), position, location, department) created at – timestamp of registration in HR department

References

https://clockster.com/about-us

https://www.w3resource.com/PostgreSQL/postgresql_unnest-function.php

https://www.hrhelpboard.com/hr-policies/attendance-policy.htm

https://wheniwork.com/blog/how-to-deal-with-employee-absenteeism

https://www.shrm.org/resourcesandtools/tools-and-samples/toolkits/pages/managingemployeeattendance.aspx