



CREATING A DASHBOARD FOR CLOCKSTER

Data Analytics Course Project 2

SUBMITTED BY:
ARSIE MAURICIO

PROJECT OVERVIEW

As part of a data analysis team within the HR department of a medical company, we are tasked with handling a substantial dataset that encompasses 10-15 parameters per day, per year, for over a thousand employees in the organization. These parameters include information related to their arrival and departure times, vacation days, sick leave, and other related data points.

The task involves responding to the following questions that have been formulated by the CEO:

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. 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.)?

DATA CLEANING AND TRANSFORMATION

This section includes the tasks performed using PostgreSQL and Power BI for data cleaning and transformation.

→ **PostgreSQL**

The following are the SQL codes used to create table and import the csv files in PostgreSQL.

ATTENDANCE

```
create table attendance (  
  user_id text,  
  first_name text,  
  last_name text,  
  location text,  
  date text,  
  time text,  
  timezone text,  
  case text,  
  source text  
)
```

LEAVE REQUESTS

```
create table leave_requests  
( user_id text,  
  first_name text,  
  last_name text,  
  type text,  
  leave_type text,  
  dates text,  
  time_start text,  
  time_end text,  
  timezone text, status text, created_at  
text );
```

PAYROLL

```
create table payroll (  
  user_id text,  
  first_name text,  
  last_name text,  
  date_start text,  
  date_end text,  
  ctc numeric,  
  net_pay numeric,  
  gross_pay numeric,  
  data_salary_basic_rate  
numeric,  
  data_salary_basic_type text,  
  currency text,  
  status text,  
  created_at text  
);
```

SCHEDULE

```
create table schedule (  
  type text,  
  dates text,  
  time_start text,  
  time_end text,  
  timezone text,  
  time_planned  
integer,  
  break_time integer,  
  leave_type text,  
  user_id text  
);
```

USERS

```
create table users (  
    user_id text,  
    first_name text,  
    last_name text,  
    gender text,  
    date_birth text,  
    date_hire text,  
    date_leave text,  
    employment  
    text,  
    position text,  
    location text,  
    department text,  
    created_at text  
);
```

After importing the datasets in PostgreSQL, we used the following SQL codes to clean the data and create updated tables in schedule and leave requests.

SCHEDULE

```
CREATE TABLE updated_schedule AS  
WITH schedules_with_row_number AS (  
    SELECT *,  
        ROW_NUMBER() OVER () AS row  
    FROM (SELECT DISTINCT * FROM schedule) sub),  
    dates_table as (select row, unnest(string_to_array(dates, ',')) as date  
        from schedules_with_row_number),  
    users_table as (select row, unnest(string_to_array(user_id, ',')) as  
user_id  
        from schedules_with_row_number)  
SELECT type,  
    time_start,  
    time_end,  
    timezone,  
    time_planned,  
    break_time,  
    leave_type,  
    dt.date,  
    ut.user_id  
from schedules_with_row_number s  
    left join dates_table dt on dt.row = s.row  
    left join users_table ut on ut.row = s.row;
```

LEAVE REQUESTS

```
CREATE TABLE updated_leave_requests AS
WITH leave_request_rownumber AS (SELECT *,
    row_number() over () AS row
    FROM leave_requests),
    dates_table AS (SELECT row, unnest(string_to_array(dates, ',')) AS
    date
        FROM leave_request_rownumber)
SELECT type,
    status,
    created_at,
    user_id,
    leave_type,
    dt.date
FROM leave_request_rownumber s
LEFT JOIN dates_table dt on dt.row = s.row;
```

→ **PowerBI**

The following are the M codes used in Power BI for additional data cleaning.

Parentheses were removed using "Replace values" tool.

UPDATED LEAVE REQUESTS

```
let
    Source = Folder.Files("C:\Arsie\Refocus\Project 2\Dataset\Final Dataset"),
    #"C:\Arsie\Refocus\Project 2\Dataset\Final Dataset\_updated_leave_requests csv1" =
    Source[{"Folder Path"="C:\Arsie\Refocus\Project 2\Dataset\Final
    Dataset\",Name="updated_leave_requests.csv"}][Content],
    #"Imported CSV" = Csv.Document(#"C:\Arsie\Refocus\Project 2\Dataset\Final
    Dataset\_updated_leave_requests csv1",[Delimiter=",", Columns=6, Encoding=1252,
    QuoteStyle=QuoteStyle.None]),
    #"Promoted Headers" = Table.PromoteHeaders(#"Imported CSV", [PromoteAllScalars=true]),
    #"Changed Type" = Table.TransformColumnTypes(#"Promoted Headers",{"type", type text},
    {"status", type text}, {"created_at", type datetime}, {"user_id", type text}, {"leave_type", type text},
    {"date", type text}},
    #"Replaced Value" = Table.ReplaceValue(#"Changed Type",[" ", " ],Replacer.ReplaceText,{"date"}),
    #"Replaced Value1" = Table.ReplaceValue(#"Replaced Value",",",",",Replacer.ReplaceText,{"date"})
in
    #"Replaced Value1"
```

UPDATED SCHEDULE

```
let
    Source = Folder.Files("C:\Arsie\Refocus\Project 2\Dataset\Final Dataset"),
    #"C:\Arsie\Refocus\Project 2\Dataset\Final Dataset\_updated_schedule csv1" = Source{[#"Folder
Path"="C:\Arsie\Refocus\Project 2\Dataset\Final
Dataset\",Name="updated_schedule.csv"]}[Content],
    #"Imported CSV" = Csv.Document(#"C:\Arsie\Refocus\Project 2\Dataset\Final
Dataset\_updated_schedule csv1",[Delimiter=";", Columns=9, Encoding=1252,
QuoteStyle=QuoteStyle.None]),
    #"Promoted Headers" = Table.PromoteHeaders(#"Imported CSV", [PromoteAllScalars=true]),
    #"Changed Type" = Table.TransformColumnTypes(#"Promoted Headers",{{"type", type text},
{"time_start", type text}, {"time_end", type text}, {"timezone", type text}, {"time_planned",
Int64.Type}, {"break_time", type text}, {"leave_type", type text}, {"date", type text}, {"user_id", type
text}}),
    #"Replaced Value" = Table.ReplaceValue(#"Changed Type",["", " ",Replacer.ReplaceText,{"date"}),
#"Replaced Value1" = Table.ReplaceValue(#"Replaced Value",",",",",Replacer.ReplaceText,{"date"}),
    #"Replaced Value2" = Table.ReplaceValue(#"Replaced
Value1",["{", ""],Replacer.ReplaceText,{"user_id"}),
    #"Replaced Value3" = Table.ReplaceValue(#"Replaced
Value2","}",",",Replacer.ReplaceText,{"user_id"}),
    #"Changed Type1" = Table.TransformColumnTypes(#"Replaced Value3",{{"date", type date},
{"user_id", Int64.Type}})
in
    #"Changed Type1"
```

Attendance and schedule tables were merged in PowerQuery using a common column (user_id&date). All other transformations were performed on the attendance table.

ATTENDANCE

```

let
    Source = Folder.Files("C:\Arsie\Refocus\Project 2\Dataset\Final Dataset"),
    #"C:\Arsie\Refocus\Project 2\Dataset\Final Dataset\_attendance csv1" = Source[{"Folder
Path"="C:\Arsie\Refocus\Project 2\Dataset\Final Dataset\",Name="attendance.csv"}][Content],
    #"Imported CSV" = Csv.Document(#"C:\Arsie\Refocus\Project 2\Dataset\Final Dataset\_attendance
csv1",[Delimiter="," , Columns=9, Encoding=1252, QuoteStyle=QuoteStyle.None]),
    #"Promoted Headers" = Table.PromoteHeaders(#"Imported CSV", [PromoteAllScalars=true]),
    #"Changed Type" = Table.TransformColumnTypes(#"Promoted Headers",{{"user_id", Int64.Type},
{"first_name", type text}, {"last_name", type text}, {"location", type text}, {"date", type date}, {"time", type
time}, {"timezone", type text}, {"case", type text}, {"source", type text}}),
    #"Merged Queries" = Table.NestedJoin(#"Changed Type", {"user_id", "date"}, updated_schedule, {"user_id",
"date"}, "updated_schedule", JoinKind.Inner),
    #"Expanded updated_schedule" = Table.ExpandTableColumn(#"Merged Queries", "updated_schedule",
{"type", "time_start", "time_end", "timezone", "time_planned", "break_time", "leave_type",
"updated_schedule.type", "updated_schedule.time_start", "updated_schedule.time_end",
"updated_schedule.timezone", "updated_schedule.time_planned", "updated_schedule.break_time",
"updated_schedule.leave_type"}),
    #"Removed Columns" = Table.RemoveColumns(#"Expanded updated_schedule",{"first_name", "last_name"}),
    #"Changed Type1" = Table.TransformColumnTypes(#"Removed Columns",{{"time", type time}}),
    #"Removed Duplicates" = Table.Distinct(#"Changed Type1"),
    #"Changed Type2" = Table.TransformColumnTypes(#"Removed Duplicates",{{"updated_schedule.time_start",
type time}, {"updated_schedule.time_end", type time}}),
    #"Inserted Time Subtraction" = Table.AddColumn(#"Changed Type2", "Subtraction", each [time] -
[updated_schedule.time_start], type duration),
    #"Inserted Time Subtraction1" = Table.AddColumn(#"Inserted Time Subtraction", "Subtraction.1", each
[updated_schedule.time_end] - [time], type duration),
    #"Inserted Total Minutes" = Table.AddColumn(#"Inserted Time Subtraction1", "Total Minutes", each
Duration.TotalMinutes([Subtraction]), type number),
    #"Inserted Total Minutes1" = Table.AddColumn(#"Inserted Total Minutes", "Total Minutes.1", each
Duration.TotalMinutes([Subtraction.1]), type number),
    #"Renamed Columns" = Table.RenameColumns(#"Inserted Total Minutes1",{{"Total Minutes", "Total
Minutes_IN"}, {"Total Minutes.1", "Total Minutes_OUT"}}),
    #"Added Conditional Column" = Table.AddColumn(#"Renamed Columns", "Minutes Late Num", each if [case]
= "IN" then [Total Minutes_IN] else if [case] = "OUT" then [Total Minutes_OUT] else 0),
    #"Added Conditional Column1" = Table.AddColumn(#"Added Conditional Column", "Late_Status", each if
[updated_schedule.type] = "leave" then "Leave" else if [updated_schedule.type] = "free" then "Leave" else if
[case] = "IN" and [Minutes Late Num]>10 then "Late" else if [case] = "OUT" and [Minutes Late Num]>0 then
"Left Early" else "On Time"),
    #"Removed Errors" = Table.RemoveRowsWithErrors(#"Added Conditional Column1", {"Late_Status"}),
    #"Filtered Rows" = Table.SelectRows(#"Removed Errors", each [Minutes Late Num] = null or [Minutes Late
Num] <= 120 and [Minutes Late Num] >= -120),
    #"Changed Type3" = Table.TransformColumnTypes(#"Filtered Rows",{{"Minutes Late Num", type number}}),
    #"Inserted Day Name" = Table.AddColumn(#"Changed Type3", "Day Name", each
Date.DayOfWeekName([date]), type text),
    #"Replaced Value" = Table.ReplaceValue(#"Inserted Day Name", "", "Not
Specified", Replacer.ReplaceValue, {"location"})
in
    #"Replaced Value"

```

Assumption: An employee is late if he or she checked in 10 minutes after the start of his or her shift and left early when he or she checked out earlier than his or her shift should have ended.

Employee positions were replaced into its English translations for better understanding.

UPDATED SCHEDULE

```
let
  Source = Folder.Files("C:\Arsie\Refocus\Project 2\Dataset\Final Dataset"),
  #"C:\Arsie\Refocus\Project 2\Dataset\Final Dataset\_users csv1" = Source[{"Folder
Path"="C:\Arsie\Refocus\Project 2\Dataset\Final Dataset\",Name="users.csv"}][Content],
  #"Imported CSV" = Csv.Document(#"C:\Arsie\Refocus\Project 2\Dataset\Final Dataset\_users
csv1",[Delimiter=";",Columns=12,Encoding=1252,QuoteStyle=QuoteStyle.None]),
  #"Promoted Headers" = Table.PromoteHeaders(#"Imported CSV",[PromoteAllScalars=true]),
  #"Changed Type" = Table.TransformColumnTypes(#"Promoted Headers",{"user_id",Int64.Type,
{"first_name",type text},{"last_name",type text},{"gender",type text},{"date_birth",type date},
{"date_hire",type date},{"date_leave",type date},{"employment",type text},{"position",type text},
{"location",type text},{"department",type text},{"created_at",type datetime})),
  #"Replaced Value" = Table.ReplaceValue(#"Changed Type","", "Not
Specified",Replacer.ReplaceValue,{"department"}),
  #"Replaced Value1" = Table.ReplaceValue(#"Replaced Value","", "Not
Specified",Replacer.ReplaceValue,{"position"}),
  #"Replaced Value2" = Table.ReplaceValue(#"Replaced Value1","Admin Bisnis","Business
Administrator",Replacer.ReplaceText,{"position"}),
  #"Replaced Value3" = Table.ReplaceValue(#"Replaced
Value2","Apoteker","Pharmacist",Replacer.ReplaceText,{"position"}),
  #"Replaced Value4" = Table.ReplaceValue(#"Replaced Value3","Asisten Apoteker","Pharmacy
Assistant",Replacer.ReplaceText,{"position"}),
  #"Replaced Value5" = Table.ReplaceValue(#"Replaced
Value4","Bidan","Midwife",Replacer.ReplaceText,{"position"}),
  #"Replaced Value6" = Table.ReplaceValue(#"Replaced Value5","Direktur Pengembangan","Director
of Development",Replacer.ReplaceText,{"position"}),
  #"Replaced Value7" = Table.ReplaceValue(#"Replaced Value6","Direktur
Utama","CEO",Replacer.ReplaceText,{"position"}),
  #"Replaced Value8" = Table.ReplaceValue(#"Replaced
Value7","Dokter","Doctor",Replacer.ReplaceText,{"position"}),
  #"Replaced Value9" = Table.ReplaceValue(#"Replaced
Value8","Perawat","Nurse",Replacer.ReplaceText,{"position"}),
  #"Replaced Value10" = Table.ReplaceValue(#"Replaced Value9","Asisten Pharmacist","Pharmacy
Assistant",Replacer.ReplaceText,{"position"})
in
  #"Replaced Value10"
```

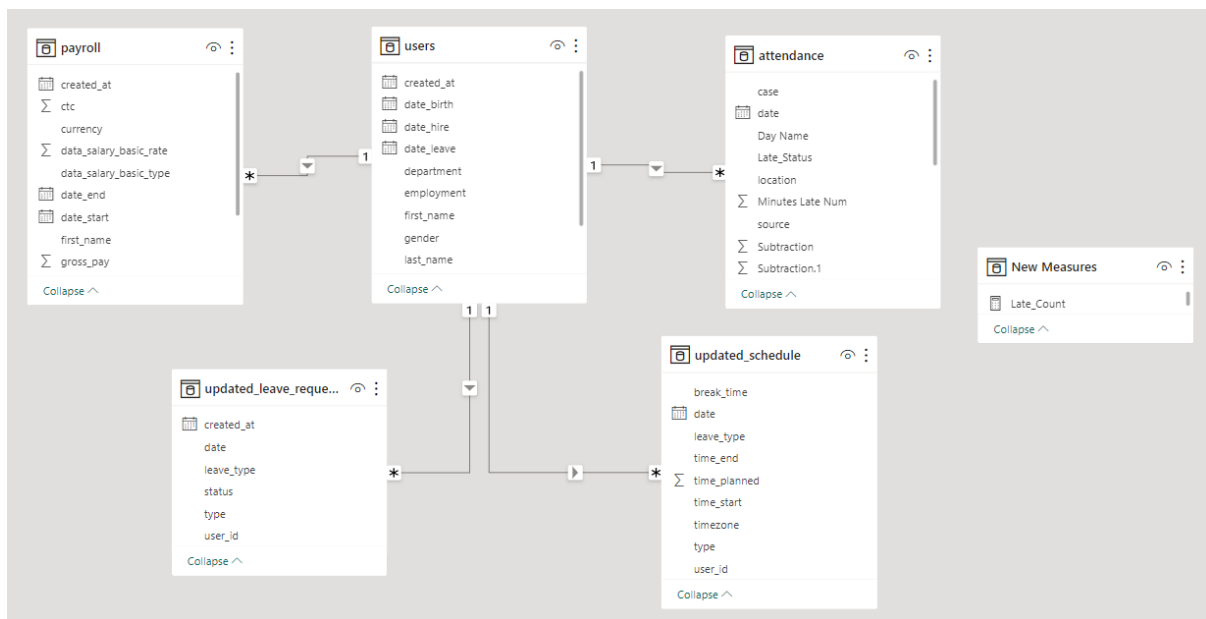

Using DAX, we calculated measures on Leave, Late and Left Early for visualization. We also used Enter Data Tool to create separate table for new measures.

DAX Formula

```
Late_Count =  
CALCULATE(COUNTROWS(attendance), attendance[Late_Status]="Late")  
Leave_Count =  
CALCULATE(COUNTROWS(attendance), attendance[Late_Status]="Leave")  
LeftEarly_Count =  
CALCULATE(COUNTROWS(attendance), attendance[Late_Status]="Left  
Early")
```

DATA MODELING

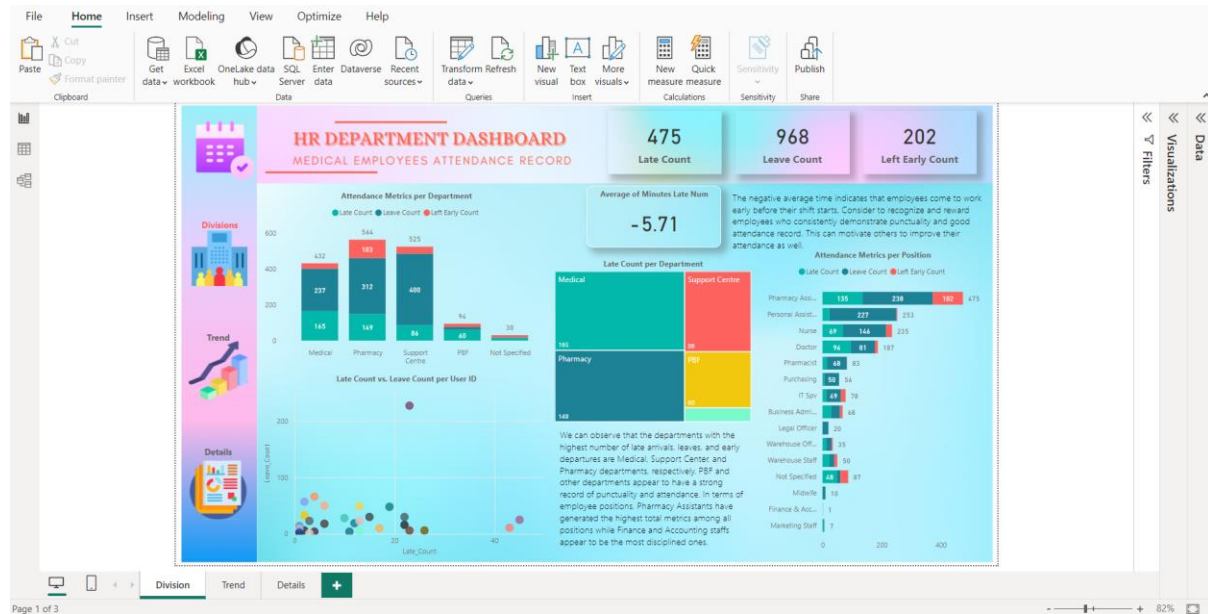
After the data cleaning and transformation, we created data model to represent the relationship between datasets. Each relationship has the correct related columns, a many-to-one cardinality, a singular cross-filter direction, and are active.



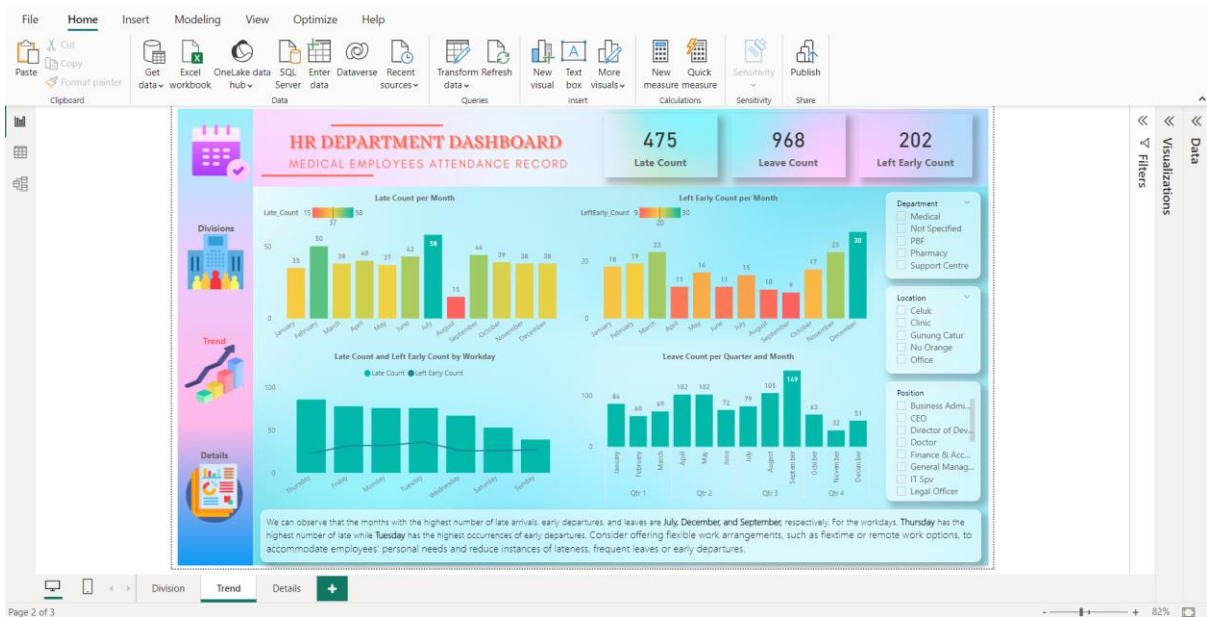
DATA VISUALIZATION, INSIGHTS, AND RECOMMENDATION

Using PowerBI, we created three Dashboards with visualization, written insights and recommendations for each question.

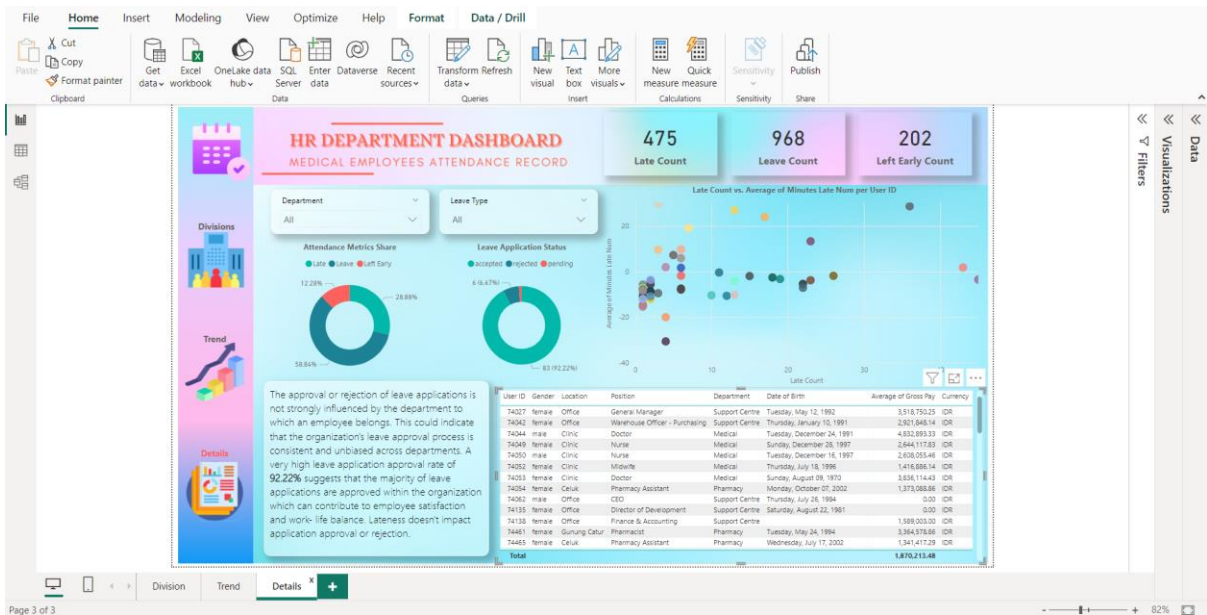
In Divisions page, we were able to identify the most disciplined and undisciplined employees and divisions.



- We can observe that the departments with the highest number of late arrivals, leaves, and early departures are Medical, Support Center, and Pharmacy departments, respectively. On the other hand, PBF and other departments appear to have a strong record of punctuality and attendance. In terms of employee positions, Pharmacy Assistants have generated the highest total metrics among all positions while Finance and Accounting staffs appear to be the most disciplined ones. We can also see here that Employee ID 83902 generated the highest number of leaves while employee ID 74054 has the highest number of late arrivals.
- However, despite the identified undisciplined divisions/employees, the negative average time of -5.71 indicates that employees in general, tend to come to work early before their shift starts which shows a positive workplace culture.
- It is recommended to recognize and reward employees who consistently demonstrate punctuality and good attendance record. This can motivate others to improve their attendance as well.



- For the attendance trend, we can observe that the months with the highest number of late arrivals, early departures, and leaves are July, December, and September, respectively. For the workdays, Thursday has the highest number of late while Tuesday has the highest occurrences of early departures.
- It is recommended to consider offering flexible work arrangements, such as flextime or remote work options, to accommodate employees' personal needs and reduce instances of lateness, frequent leaves or early departures.



- Lastly, for the other details of employees' records, we can observe that the approval or rejection of leave applications is not strongly influenced by the department to which an employee belongs. This could indicate that the organization's leave approval process is consistent and unbiased across departments.
- A very high leave application approval rate of 92.22% suggests that the majority of leave applications are approved within the organization which can contribute to employee satisfaction and work-life balance. Lateness doesn't impact application approval or rejection.

- It is recommended to continuously monitor these trends, approval rates, and gather feedback from employees and managers. This ongoing analysis can help identify any emerging issues and allow for timely adjustments to policies and practices.