In [1]:
```
## Importing required libraries

from sqlalchemy import create_engine
import pandas as pd
import datetime as dt

import mysql.connector
import warnings
warnings.filterwarnings("ignore")
```

In [2]:
```
## accessing database password using json so that it is hidden

import json

# Specify the file path of the JSON file
file_path = r"C:\Users\Admin\Downloads\Hr_Analytics Document'\database_data.json"

# Read data from the JSON file
with open(file_path, 'r') as json_file:
    data = json.load(json_file)

# Access information from the loaded JSON data
db_password = data['database Password']
```

In [3]:
```
## Reading excel file
attentence_df= pd.read_excel(r"C:\Users\Admin\Downloads\Hr_Analytics Document'\Attendance Sheet 2022-2023_Masked.xlsx",
                             sheet_name = None)
```

```
                                sheet_name = None)
```

In [4]:
```
## accessing each sheet form file

apr_attendence_df = attentence_df['Apr 2022']
may_attendence_df = attentence_df['May 2022']
june_attendence_df = attentence_df['June 2022']
```

In [5]:
```
may_attendence_df.head(2)
```

Out[5]:

| | AtliQ | Unnamed: 1 | 2022-05-02 00:00:00 | 2022-05-03 00:00:00 | 2022-05-04 00:00:00 | 2022-05-05 00:00:00 | 2022-05-06 00:00:00 | 2022-05-07 00:00:00 | 2022-05-08 00:00:00 | 2022-05-09 00:00:00 | ... | Work from home | Paid Leave | Sick Leave | Birthday Leave | Floting festival leave | Bereavement Leave |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Employee Code | Name | 2022-05-02 00:00:00 | 2022-05-03 00:00:00 | 2022-05-04 00:00:00 | 2022-05-05 00:00:00 | 2022-05-06 00:00:00 | 2022-05-07 00:00:00 | 2022-05-08 00:00:00 | 2022-05-09 00:00:00 | ... | WFH | PL | SL | BL | FFL | BRL |
| 1 | Atq -406 | Thanos Thakur | P | P | P | P | P | WO | WO | P | ... | 0 | 0 | 0 | 0 | 0 | 0 |

2 rows × 45 columns

In [6]:
```
## Data Transformation Code

def transform_data(df):
    column_nam = df.iloc[0].values
    df =df.iloc[1: ,:]

    df.columns = column_nam
    df = df.rename(columns={df.columns[0]: 'Emp_Code',
                            df.columns[1]: 'Emp_Name'})

    df = pd.melt(df, id_vars=['Emp_Code', 'Emp_Name'],
                        var_name='Date')
    df['Date'] = pd.to_datetime(df['Date'] ,errors='coerce')
    df = df.dropna(subset = ['Date'])

    current_month_name = df['Date'].dt.month_name().value_counts()
    max_value = current_month_name.max()
    current_month_name = current_month_name[current_month_name == max_value].index[0]
    df = df[(df['Date'].dt.month_name()) == current_month_name]
    df.rename(columns ={'value':'Attendence'},inplace = True)
    return df
```

In [7]:
```
apr_attendence_df = transform_data(apr_attendence_df)
may_attendence_df = transform_data(may_attendence_df)
june_attendence_df = transform_data(june_attendence_df)
```

In [7]:
```
apr_attendence_df = transform_data(apr_attendence_df)
may_attendence_df = transform_data(may_attendence_df)
june_attendence_df = transform_data(june_attendence_df)
```

In [8]:
```
## Transformed Data and checking wheather data is correct

may_attendence_df.head()
```

Out[8]:

| | Emp_Code | Emp_Name | Date | Attendence |
|---|---|---|---|---|
| **0** | Atq -406 | Thanos Thakur | 2022-05-02 | P |
| **1** | Atq -462 | Jarvis Singh | 2022-05-02 | P |
| **2** | Atq -411 | Nevaeh Waller | 2022-05-02 | P |
| **3** | Atq -398 | Bo Cordova | 2022-05-02 | P |
| **4** | Atq -438 | Mekhi Singleton | 2022-05-02 | P |

In [10]:
```
## Appending all Sheets together

all_record_df = pd.concat([apr_attendence_df ,may_attendence_df ,june_attendence_df ]
                          ,axis = 0)
all_record_df.shape
```

Out[10]: (7493, 4)

Out[10]: (7499, 4)

In [12]:
```
## connection

connection_url = f'mysql+mysqlconnector://root:{db_password}@localhost:3306/Hr_Analytics'

# Create an SQLAlchemy engine
engine = create_engine(connection_url)

# Load DataFrame into MySQL
all_record_df.to_sql('Emp_Attendence_Data', con=engine, if_exists='append', index=False)

# Close the connection
engine.dispose()
```

In [ ]:

```sql
-- Designing a database for Employee Attendence Data


CREATE DATABASE Hr_Analytics ;
CREATE TABLE Hr_Analytics.Emp_Attendence_Data (
        Emp_Code VARCHAR(15) ,
        Emp_Name VARCHAR(50),
        Date DATE,
        Attendence VARCHAR(15)
    ) ;

 -- to check
select count(*) from  Hr_Analytics.Emp_Attendence_Data;
```

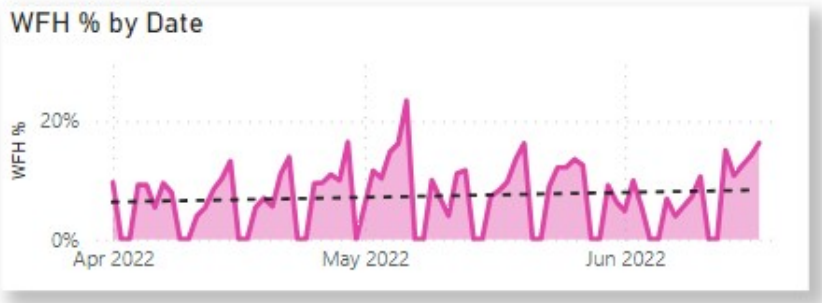
# HR ANALYTICS DASHBOARD

Apr 22 | May 22 | Jun 22

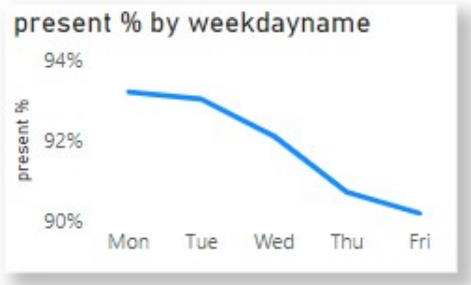
| 91.83% | 10.00% | 1.10% |
|---|---|---|
| present % | WFH % | SL % |

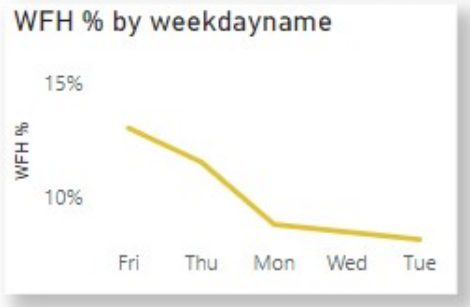
| Emp_Name | present % | WFH % | SL % |
|---|---|---|---|
| Alexander Davenport | 100.00% | 100.00% | 0.00% |
| Alyson Huber | 100.00% | 0.00% | 0.00% |
| Charity Singleton | 100.00% | 0.00% | 0.00% |
| Ciara Allison | 100.00% | 0.00% | 0.00% |
| Greta Horton | 100.00% | 4.76% | 0.00% |
| Gustavo Ritter | 100.00% | 100.00% | 0.00% |
| Isabella Pittman | 100.00% | 0.00% | 0.00% |
| Isiah Small | 100.00% | 0.00% | 0.00% |
| **Total** | **91.83%** | **10.00%** | **1.10%** |

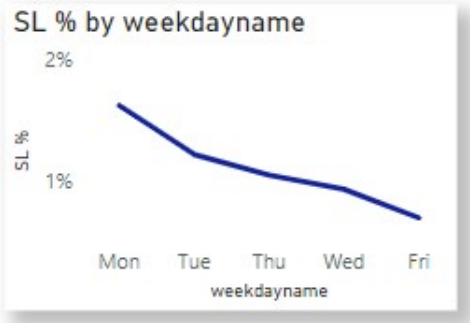
| Emp_Name | Monday, April 04, 2022 | Tuesday, April 05, |
|---|---|---|
| Titus Andersen | P | P |
| Tori Shannon | P | P |
| Trystan Ortega | P | P |
| Tucker Austin | P | P |
| Weston Horton | P | P |
| Will Mahoney | P | P |
| Xiomara Ruiz | P | P |
| Zaiden Wheeler | | |
| **Total** | **HPL** | **HPL** |






**WFH : Work From Home** **SL : Sick Leave**