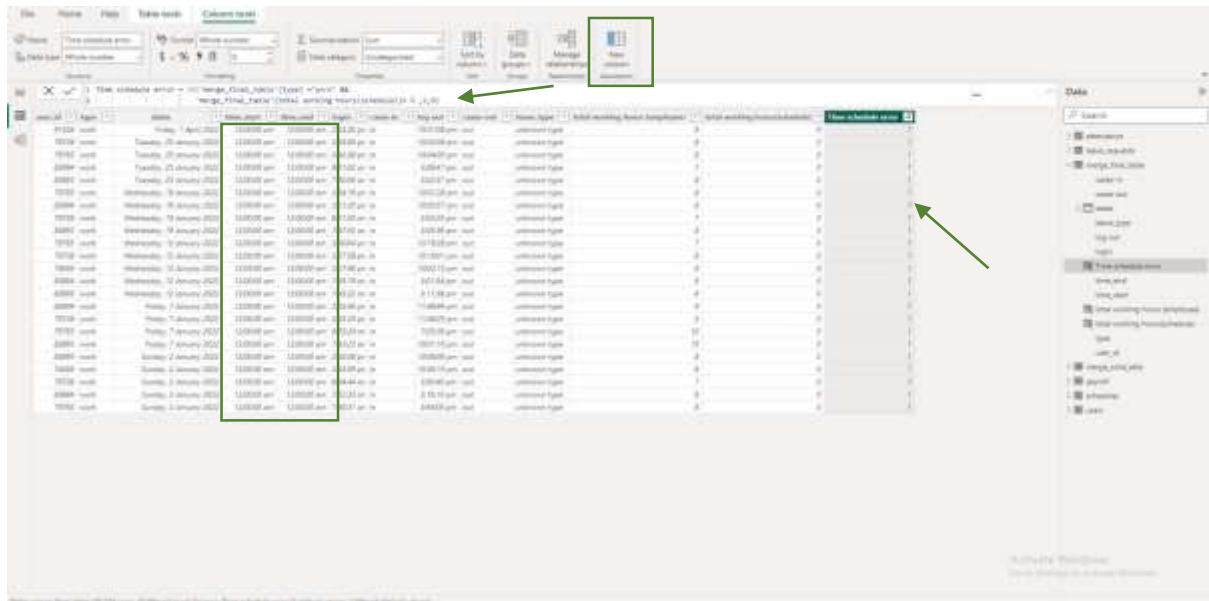


In the Data View of merge_final_table, creating column for Time schedule error. Click New Column and type the DAX code,

Time schedule error = if('merge_final_table'[type] = "work" &&
'merge_final_table'[total working hours(schedule)] = 0 ,1,0)

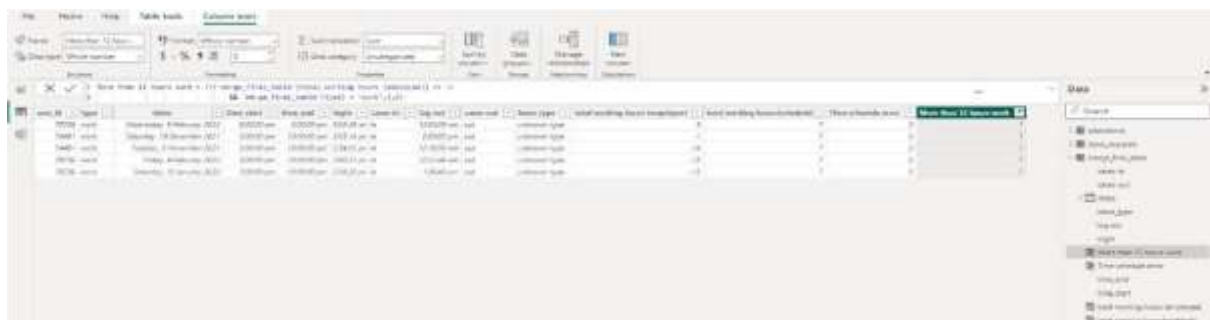
This code will filter the time schedule of the employees, the start time and end time with the same time value. The type is equal to “work”. In the column Time schedule error “1” means “yes” and “0” means “no”.



In the Data View of merge_final_table, creating column for More than 12-hour work. Click New Column and type the DAX code,

More than 12 hours work = if ('merge_final_table'[total working hours (employee)] <= -1
&& 'merge_final_table'[type] = "work",1,0)

This code will filter the employees that work more than 12 hours from their schedule.



login vs start time = DATEDIFF('merge_final_table'[time_start],'merge_final_table'[login], minute)

[illegible]

```
log out vs end time) = DATEDIFF('merge_final_table'[time_end], 'merge_final_table'[log out],
MINUTE)
```

This column will serve as our reference to locate the log out, done by the user. The value 0 means the user log out right on time, when the value is -1 and below means the user log out early and when the value is 1 and above, it means the user log out exceeding the time schedule.

The screenshot shows a data table with multiple columns. The 'Log out as user' column is highlighted with a green box. Two green arrows point to specific rows in the table, indicating data points of interest.

We also use the date hire column from the user table as filter for our merge_final_table. This column will serve us as our basis to know only what date needs to be included in our data analysis. We update the date hire column because it has blank values in it. We use the payroll tables start date as our reference so that we can put some value in the blank cells of date hire.

In the Data View of users, Click the column date_hire go to Column tools and set the Data type to text.

The screenshot shows a data table in a software interface. A green box highlights the 'Column tools' button. A green arrow points to the 'date_hire' column. Another green arrow points to the 'Data type' dropdown menu.

Create column for date hire update. Click New Column and type the DAX code, date_hire

update = SWITCH(TRUE(),

'users'[user_id] = 74008, "3/1/2022", 'users'[user_id] = 74025, "11/1/2021",

'users'[user_id] = 74062, "3/1/22", 'users'[user_id] = 74135, "3/1/22", 'users'[user_id] = 74138, "11/1/21",

'users'[user_id] = 74745, "3/1/22", 'users'[user_id] = 75218, "11/1/21", 'users'[user_id]
= 75834, "11/1/21",

'users'[user_id] = 75848, "11/1/21", 'users'[user_id] = 75955, "11/1/21", 'users'[user_id]
= 75963, "11/1/21",

'users'[user_id] = 75983, "11/1/21", 'users'[user_id] = 75986, "11/1/21", 'users'[user_id]
= 81587, "3/1/22",

'users'[user_id] = 84490, "12/1/21", 'users'[user_id] = 84509, "12/1/21", 'users'[user_id]
= 84517, "12/1/21",

'users'[user_id] = 84699, "12/1/21", 'users'[user_id] = 84932, "12/1/21", 'users'[user_id]
= 85877, "12/1/21",

'users'[user_id] = 90377, "12/1/21", 'users'[user_id] = 93607, "2/1/22", 'users'[user_id]
= 120694, "4/1/22",

'users'[user_id] = 120696, "4/1/22", 'users'[user_id] = 125721, "4/1/22", 'users'[user_id]
= 125744, "4/1/22",

'users'[user_id] = 132484, "8/1/22", 'users'[user_id] = 75914, "11/1/21", 'users'[user_id]
= 125748, "4/1/22",

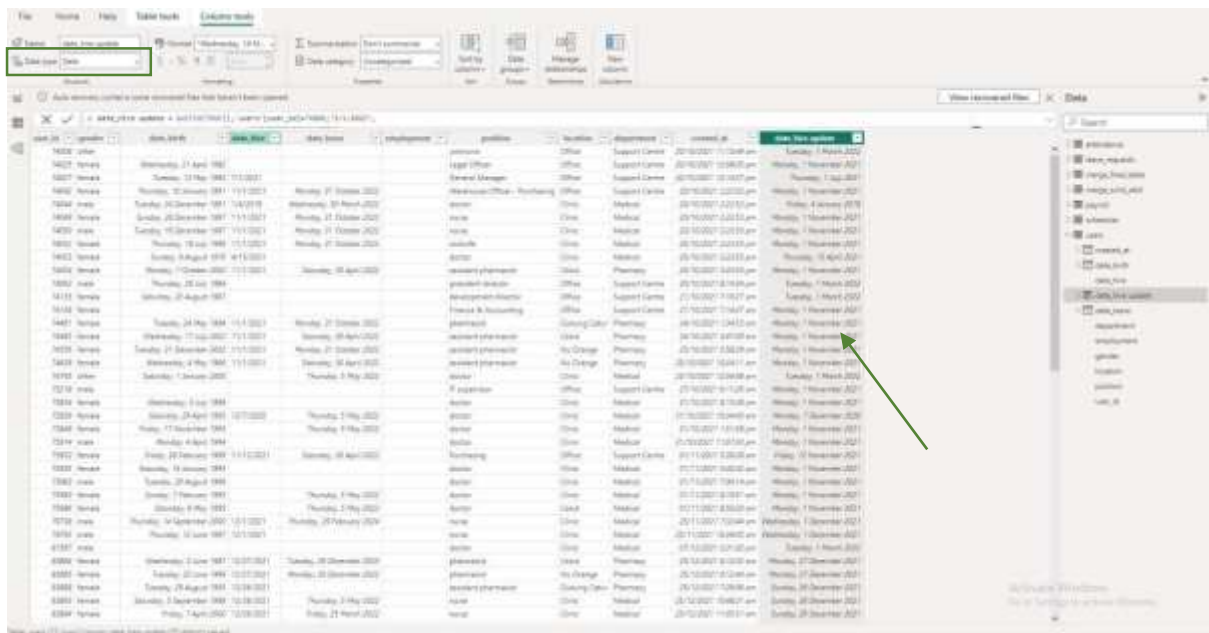
'users'[user_id] = 126082, "4/1/22", 'users'[user_id] = 129675, "4/1/22", 'users'[user_id]
= 130650, "5/1/22",

'users'[user_id] = 157837, "10/1/22", 'users'[user_id] = 157916, "10/1/22",

'users'[user_id] = 159207, "11/1/22", 'users'[user_id] = 159217, "11/1/22",

'users'[user_id] = 160306, "11/1/22", 'users'[user_id] = 84488, "12/1/21", 'users'[date_hire])

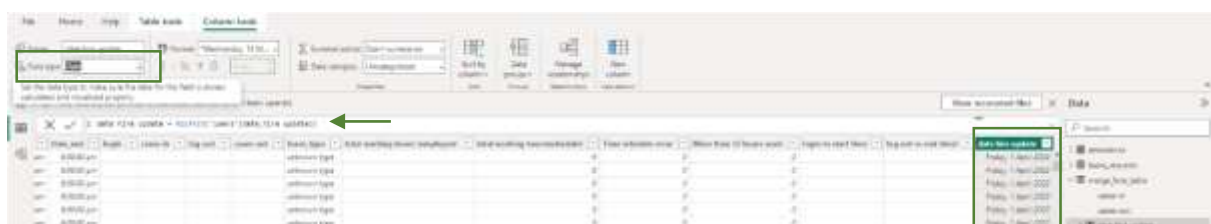
After typing the code, the new column for date_hire update is created. Set the data type of date_hire update column to date. Next is to get the date_hire update data into the merge_final_table.



In the Data View of merge_final_table, create a column for date hire update. Click New Column and type the DAX code,

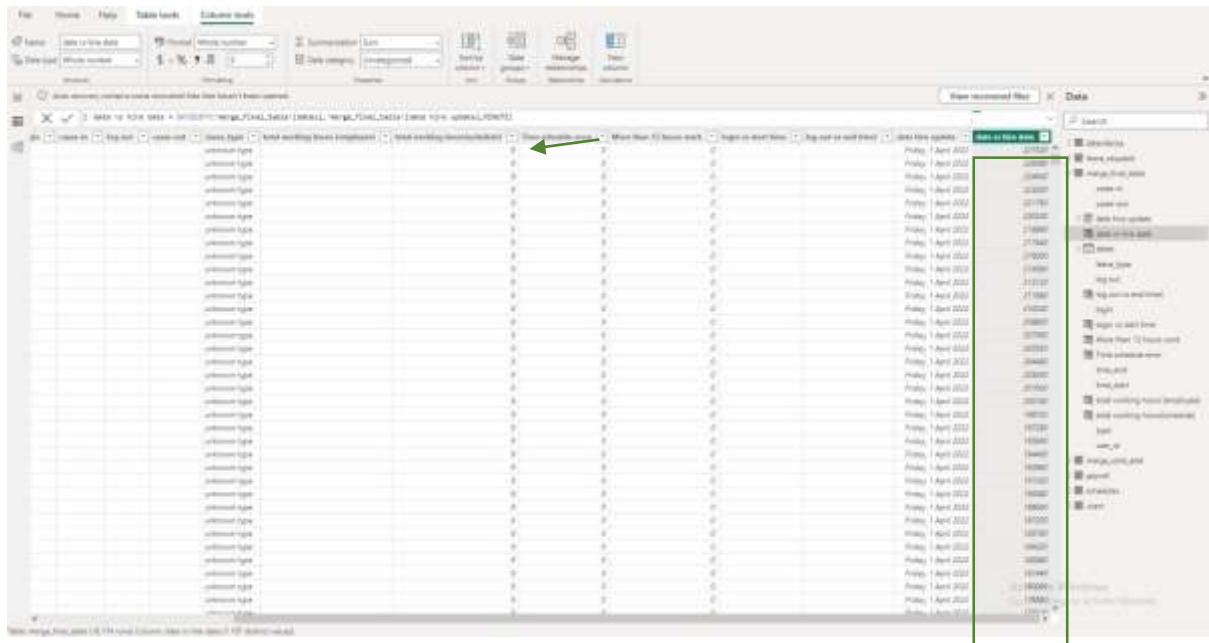
date hire update = RELATED ('users'[date_hire update])

This code call the value of date_hire update to appear in the merge_final_table.



To know if the user is hired or not yet hire, create a column named date vs date hire. This column will measure the difference between dates vs date hire update. When value of date vs hire is 0 and below it means the user is hired. When the value is 1440 and above, this means that the user has not been hired yet. Click New Column and type the DAX code,

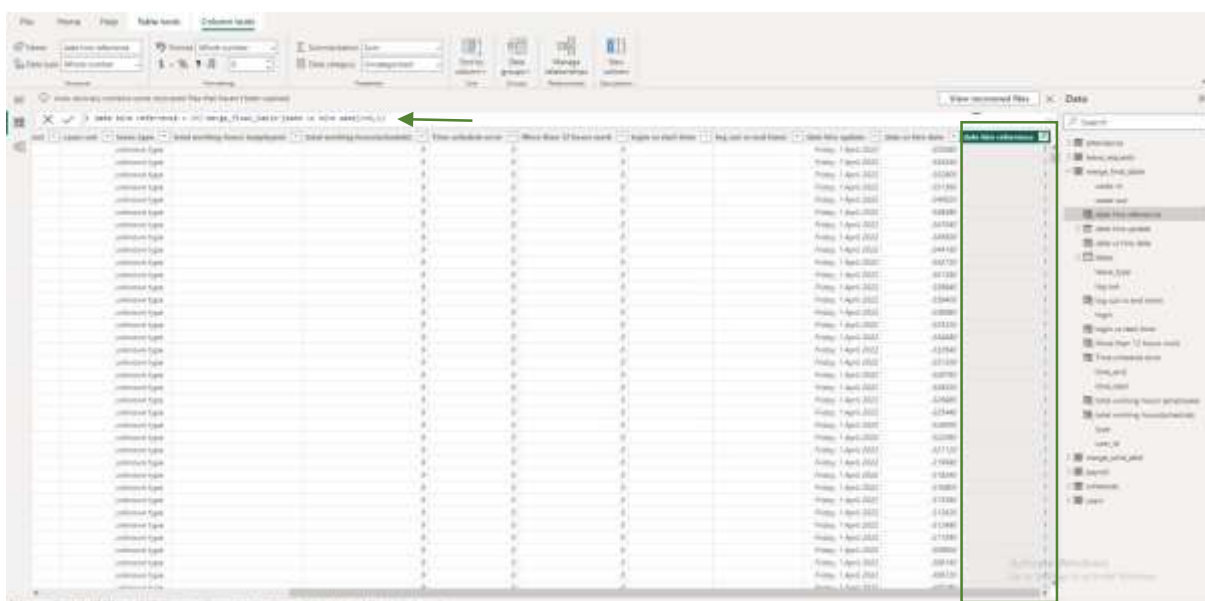
date vs hire date = DATEDIFF('merge_final_table'[dates],'merge_final_table'[date hire update],MINUTE)



In the Data View of merge_final_table, create a column for date hire reference. Click New Column and type the DAX code,

date hire reference = if('merge_final_table'[date vs hire date]<=0,1)

There are too many values in the date vs hire date column. This code will make it easier to access the value of date vs hire date. When the value of date hire reference is 1 it means “yes” the user is hired. The date hire update column will show the actual date.



[illegible]

In the Data View of merge_final_table, create a column for Not Late (regular schedule). Click New Column and type the DAX code,

Not Late (regular schedule) = if('merge_final_table'[type] = "work" &&

'merge_final_table'[login vs start time] in {10,9,8,7,6,5,4,3,2,1,0,-1,-2,-3,-4,-5,-6,-7,-8,-9,-10,-11,-12,-13,-14,-15,-16,-17,-18,-19,-20,-21,-22,-23,-24,-25,-26,-27,-28,-29,-30,-31,-32,-33,-34,-35,-36,-37,-38,-39,-40,-41,-42,-43,-44,-45,-46,-47,-48,-49,-50,-51,-52,-53,-54,-55,-56,-57,-58,-59} &&

'merge_final_table'[log out vs end time] in {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59} &&

'merge_final_table'[date hire reference]=1 &&

'merge_final_table'[user date vs date]=0&&

'merge_final_table'[login] <> BLANK() ,1,0)

This column shows that the user has log in an hour before his/her schedule time. If not, the user has 10-minute grace period to be safe for being late. This column also shows that the user has logged out exactly or beyond the set log out, but the log out should not exceed one hour.

The screenshot displays the Power BI Desktop interface with the 'Data View' of a table named 'merge_final_table'. The table contains the following columns: ID, Name, Type, Time_Start, Time_End, Login, Log_Out, Login vs Start Time, Log Out vs End Time, Total Working Hours Category, Total Working Hours Available, Total Working Hours, and Login vs Start Time. The data is filtered by 'Type' = 'work'. The 'Login vs Start Time' column shows values ranging from -10 to 59, and the 'Log Out vs End Time' column shows values ranging from 0 to 59. The 'Total Working Hours Category' column shows values like 'Not Late (regular schedule)' and 'Late (regular schedule)'. The 'Total Working Hours Available' column shows values like '1.0' and '0.0'. The 'Total Working Hours' column shows values like '1.0' and '0.0'. The 'Login vs Start Time' column shows values like '1.0' and '0.0'.

In the Data View of merge_final_table, create a column for Late (regular schedule). Click New Column and type the DAX code,

Late (regular schedule) = if('merge_final_table'[type] = "work" &&

'merge_final_table'[date hire reference]=1 &&

'merge_final_table'[user date vs date]=0&&

'merge_final_table'[login vs start time]

in {11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59} &&

'merge_final_table'[log out vs end time] in

{0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59} &&

'merge_final_table'[login] <> BLANK(),1,0)

The column Late (regular schedule) shows that the user failed to log in on time and he/she also exceeds the 10-minute grace period to log in. The column also shows that the user has logged out exactly or beyond the specified logout, the logout must not exceed one hour.

Result of Late(regular schedule)

The screenshot displays the 'Data View' of a table named 'merge_final_table'. The 'Columns' tab is active, showing the formula for the 'Late (regular schedule)' column: `=if('merge_final_table'[type] = "work" && 'merge_final_table'[date hire reference]=1 && 'merge_final_table'[user date vs date]=0 && 'merge_final_table'[login vs start time] in {11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59} && 'merge_final_table'[log out vs end time] in {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59} && 'merge_final_table'[login] <> BLANK(),1,0)`. The table data includes columns for ID, Name, Type, Date, Time, and various time-related metrics. The 'Late (regular schedule)' column is highlighted, showing values of 1 (indicating late) and 0 (indicating on time).

In the Data View of merge_final_table, create column for Not Late (regular schedule) with over time. Click New Column and type the DAX code,

Not Late(regular schedule) with over time = if('merge_final_table'[type] = "work" &&

'merge_final_table'[login vs start time] in {10,9,8,7,6,5,4,3,2,1,0,-1,-2,-3,-4,-5,-6,-7,-8,-9,-10,-11,-12,-13,-14,-15,-16,-17,-18,-19,-20,-21,-22,-23,-24,-25,-26,-27,-28,-29,-30,-31,-32,-33,-34,-35,-36,-37,-38,-39,-40,-41,-42,-43,-44,-45,-46,-47,-48,-49,-50,-51,-52,-53,-54,-55,-56,-57,-58,-59} &&

'merge_final_table'[log out vs end time] >= 60 &&

'merge_final_table'[total working hours (employee)]>'merge_final_table'[total working hours(schedule)]&&

'merge_final_table'[date hire reference]=1 &&

'merge_final_table'[user date vs date]=0&&

'merge_final_table'[login] <> BLANK(),1,0)

The column of Not Late (regular schedule) with over time shows that the user log in less than one hour before the time schedule of logging in, if the user exceeds the time schedule, the user has a 10-minute grace period to log in. The user logged out after an hour or more.

Result of Not Late (regular schedule) with over time

Employee ID	Login Time	End Time	Not Late (regular schedule) with over time
00001	08:00:00	17:00:00	1
00002	08:00:00	17:00:00	1
00003	08:00:00	17:00:00	1
00004	08:00:00	17:00:00	1
00005	08:00:00	17:00:00	1
00006	08:00:00	17:00:00	1
00007	08:00:00	17:00:00	1
00008	08:00:00	17:00:00	1
00009	08:00:00	17:00:00	1
00010	08:00:00	17:00:00	1
00011	08:00:00	17:00:00	1
00012	08:00:00	17:00:00	1
00013	08:00:00	17:00:00	1
00014	08:00:00	17:00:00	1
00015	08:00:00	17:00:00	1
00016	08:00:00	17:00:00	1
00017	08:00:00	17:00:00	1
00018	08:00:00	17:00:00	1
00019	08:00:00	17:00:00	1
00020	08:00:00	17:00:00	1
00021	08:00:00	17:00:00	1
00022	08:00:00	17:00:00	1
00023	08:00:00	17:00:00	1
00024	08:00:00	17:00:00	1
00025	08:00:00	17:00:00	1
00026	08:00:00	17:00:00	1
00027	08:00:00	17:00:00	1
00028	08:00:00	17:00:00	1
00029	08:00:00	17:00:00	1
00030	08:00:00	17:00:00	1
00031	08:00:00	17:00:00	1
00032	08:00:00	17:00:00	1
00033	08:00:00	17:00:00	1
00034	08:00:00	17:00:00	1
00035	08:00:00	17:00:00	1
00036	08:00:00	17:00:00	1
00037	08:00:00	17:00:00	1
00038	08:00:00	17:00:00	1
00039	08:00:00	17:00:00	1
00040	08:00:00	17:00:00	1
00041	08:00:00	17:00:00	1
00042	08:00:00	17:00:00	1
00043	08:00:00	17:00:00	1
00044	08:00:00	17:00:00	1
00045	08:00:00	17:00:00	1
00046	08:00:00	17:00:00	1
00047	08:00:00	17:00:00	1
00048	08:00:00	17:00:00	1
00049	08:00:00	17:00:00	1
00050	08:00:00	17:00:00	1

In the Data View of merge_final_table, create a column for Late (regular schedule) with overtime. Click New Column and type the DAX code,

```
Late(regular schedule) with overtime = if( 'merge_final_table'[type] = "work" &&
'merge_final_table'[login vs start time]
in {11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,
42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59} &&
'merge_final_table'[log out vs end time] >=60 &&
'merge_final_table'[date hire reference]=1 &&
'merge_final_table'[total working hours (employee)]>'merge_final_table'[total working
hours(schedule)]&&
'merge_final_table'[user date vs date]=0&&
'merge_final_table'[login] <> BLANK() ,1,0)
```

The Late (regular schedule) with overtime column shows that the user log in late and exceed the 10-minute grace period. The user logged out after an hour or more.

Result of Late (regular schedule) with overtime

emp_id	type	login	log out	total working hours (employee)	total working hours (schedule)	late (regular schedule) with overtime
100000	work	10/10/2022	10/10/2022	8:00:00 am - 5:00:00 pm	8:00:00 am - 5:00:00 pm	0
100001	work	10/10/2022	10/10/2022	8:00:00 am - 5:00:00 pm	8:00:00 am - 5:00:00 pm	0
100002	work	10/10/2022	10/10/2022	8:00:00 am - 5:00:00 pm	8:00:00 am - 5:00:00 pm	0
100003	work	10/10/2022	10/10/2022	8:00:00 am - 5:00:00 pm	8:00:00 am - 5:00:00 pm	0
100004	work	10/10/2022	10/10/2022	8:00:00 am - 5:00:00 pm	8:00:00 am - 5:00:00 pm	0
100005	work	10/10/2022	10/10/2022	8:00:00 am - 5:00:00 pm	8:00:00 am - 5:00:00 pm	0
100006	work	10/10/2022	10/10/2022	8:00:00 am - 5:00:00 pm	8:00:00 am - 5:00:00 pm	0
100007	work	10/10/2022	10/10/2022	8:00:00 am - 5:00:00 pm	8:00:00 am - 5:00:00 pm	0
100008	work	10/10/2022	10/10/2022	8:00:00 am - 5:00:00 pm	8:00:00 am - 5:00:00 pm	0
100009	work	10/10/2022	10/10/2022	8:00:00 am - 5:00:00 pm	8:00:00 am - 5:00:00 pm	0

In the Data View of merge_final_table, create a column for Not Late (regular schedule) but undertime. Click New Column and type the DAX code,

Not Late (regular schedule) but undertime = if('merge_final_table'[type] = "work" &&

'merge_final_table'[login vs start time] in {10,9,8,7,6,5,4,3,2,1,0,-1,-2,-3,-4,-5,-6,-7,-8,-9,-10,-11,-12,-13,-14,-15,-16,-17,-18,-19,-20,-21,-22,-23,-24,-25,-26,-27,-28,-29,-30,-31,-32,-33,-34,-35,-36,-37,-38,-39,-40,-41,-42,-43,-44,-45,-46,-47,-48,-49,-50,-51,-52,-53,-54,-55,-56,-57,-58,-59} &&

'merge_final_table'[log out vs end time] <=-1 &&

'merge_final_table'[date hire reference]=1 &&

'merge_final_table'[More than 12 hours work] <> 1 &&

'merge_final_table'[total working hours (employee)]<='merge_final_table'[total working hours(schedule)] &&

'merge_final_table'[user date vs date]=0&&

'merge_final_table'[login] <> BLANK() ,1,0)

The Not Late (regular schedule) but undertime column shows that the user log in exactly or less than one hour before the time schedule. The user has 10-minute grace period if he/she exceed the given time to log in. This column also shows that the user logged out earlier than his/her schedule time.

Not Late(regular schedule) but undertime result.

File

Home

Insert

Table Tools

Columns Tools

Connections

Blank

Table template

Formulas

Blank workbook

Connections

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

Table template

Table

[illegible]

```
'merge_final_table'[login] <> BLANK() ,1,0)
```

The column change schedule(early login with overtime) shows that the user changed his/her login time, he/she logged earlier an hour or more than the given schedule time. This column also shows the user has logged out after an hour or more of his/her schedule time.

change schedule(early login with overtime) result.

[illegible]

```
change schedule(early login with undertime) = if( 'merge_final_table'[type] = "work" &&
    'merge_final_table'[login vs start time]<= -60 &&
    'merge_final_table'[date hire reference]=1 &&
    'merge_final_table'[user date vs date]=0&&
    'merge_final_table'[change schedule(early login with complete work schedule)]=0&&
    'merge_final_table'[total working hours (employee)]<'merge_final_table'[total working
hours(schedule)] &&
    'merge_final_table'[login] <> BLANK() ,1,0)
```

[illegible]

In the Data View of merge_final_table, create column for change schedule(late login regular log out). Click New Column and type the DAX code,

change schedule (late login regular logout) = if('merge_final_table'[type] = "work" &&

'merge_final_table'[date hire reference]=1 &&

'merge_final_table'[user date vs date]=0&&

'merge_final_table'[login vs start time] >= 60 &&

'merge_final_table'[total working hours (employee)] >'merge_final_table'[total working hours(schedule)] &&

'merge_final_table'[Time schedule error]<> 1&&

'merge_final_table'[login] <> BLANK() ,1,0)

This column shows that the user changed his/her schedule and log in late, after an hour or more from the schedule time. This column also show that the user has completed the schedule time he/she has.

The screenshot shows the Power BI Data View interface. The ribbon at the top includes 'Home', 'Modeling', 'Visuals', 'Fields', and 'Tools'. The 'Fields' pane on the right shows a list of columns, with 'change schedule' selected. The main table view displays data for various employees, with columns including 'Employee ID', 'Date', 'Type', 'Login', 'Logout', 'Total working hours (employee)', 'Total working hours (schedule)', 'Time schedule error', and 'change schedule'. The 'change schedule' column contains values of 1 or 0, indicating whether the employee's schedule was changed.

Employee ID	Date	Type	Login	Logout	Total working hours (employee)	Total working hours (schedule)	Time schedule error	change schedule
00001	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00002	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00003	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00004	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00005	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00006	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00007	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00008	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00009	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00010	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00011	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00012	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00013	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00014	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00015	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00016	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00017	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00018	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00019	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00020	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00021	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00022	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00023	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00024	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00025	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00026	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00027	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00028	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00029	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00030	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00031	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00032	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00033	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00034	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00035	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00036	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00037	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00038	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00039	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00040	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00041	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00042	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00043	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00044	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00045	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00046	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00047	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00048	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00049	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00050	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00051	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00052	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00053	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00054	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00055	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00056	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00057	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00058	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00059	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00060	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00061	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00062	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00063	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00064	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00065	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00066	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00067	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00068	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00069	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00070	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00071	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00072	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00073	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00074	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00075	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00076	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00077	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00078	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00079	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00080	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00081	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00082	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00083	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00084	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00085	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00086	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00087	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00088	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00089	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00090	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00091	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00092	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00093	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00094	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00095	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00096	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00097	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00098	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00099	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0
00100	10/10/2022	work	08:00:00	17:00:00	9.0000	9.0000	0	0

In the Data View of merge_final_table, create a column for change schedule(late login with overtime). Click New Column and type the DAX code,

```
change schedule (late login with overtime) = if( 'merge_final_table'[type] = "work" &&
'merge_final_table'[login vs start time] >= 60 &&
'merge_final_table'[date hire reference]=1 &&
'merge_final_table'[user date vs date]=0&&
'merge_final_table'[total working hours (employee)] in {10} &&
'merge_final_table'[total working hours(schedule)] in {7} &&
'merge_final_table'[user_id] in {74461,83884,74465}&&
'merge_final_table'[Time schedule error] <> 1 &&
'merge_final_table'[total working hours (employee)] >'merge_final_table'[total working
hours(schedule)] &&
'merge_final_table'[login] <> BLANK() ,1,0)
```

This column shows that the user changed his/her schedule and log in late, after an hour or more from the schedule time. This column also show that the user logs out and managed to work over time.

user_id	date	time	login	change schedule (late login with overtime)
74461	10/01/2022	08:00:00	08:00:00	0
83884	10/01/2022	08:00:00	08:00:00	0
74465	10/01/2022	08:00:00	08:00:00	0

In the Data View of merge_final_table, create a column for change schedule(late login with undertime). Click New Column and type the DAX code,

```
change schedule (late login and undertime) = if( 'merge_final_table'[type] = "work" &&
'merge_final_table'[login vs start time] >= 60 &&
'merge_final_table'[date hire reference]=1 &&
'merge_final_table'[user date vs date]=0&&
'merge_final_table'[More than 12 hours work] <> 1 &&
'merge_final_table'[Time schedule error] <> 1 &&
'merge_final_table'[total working hours (employee)] <='merge_final_table'[total working
hours(schedule)] &&
'merge_final_table'[login] <> BLANK() ,1,0)
```

This column shows that the user changed his/her schedule and log in late, after an hour or more from the schedule time. This column also shows that the user logs out early than his/her working schedule.

FileHomeSettingsToolsTools

Undo

Redo

Find

Print

Print Range

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

Print Comments

Print Formulas

Print Tables

Print Charts

Print All

Print Selection

In the Data View of merge_final_table, create a column for user obedience category. Click New Column and type the DAX code,

```
user obedience category = if('merge_final_table'[date hire reference]= 1 ,
    if('merge_final_table'[absent] = 1,"undiscipline",
    if('merge_final_table'[Not Late (regular schedule)]=1,"discipline",
    if('merge_final_table'[Late (regular schedule)]= 1,"undiscipline",
    if('merge_final_table'[Not Late(regular schedule) with over time]= 1,"discipline",
    if('merge_final_table'[Late(regular schedule) with overtime]=1,"undiscipline",
    if('merge_final_table'[Not Late (regular schedule) but undertime]=1,"undiscipline",
    if('merge_final_table'[Late (regular schedule) and undertime]=1,"undiscipline",
    if('merge_final_table'[change schedule(early login with complete work
schedule)]=1,"discipline",
    if('merge_final_table'[change schedule(early login with overtime)]=1,"discipline",
    if('merge_final_table'[change schedule(early login with undertime)]=1,"undiscipline",
    if('merge_final_table'[change schedule (late login regular logout)]=1,"undiscipline",
    if('merge_final_table'[change schedule (late login with overtime)]=1,"undiscipline",
    if('merge_final_table'[change schedule (late login and
undertime)]=1,"undiscipline"
))))))))))
```

This column shows the user's categorization whether he or she is disciplined or undisciplined user.

The screenshot shows the Microsoft Excel interface with the following details:

- Formula Bar:** Contains the DAX formula: `user obedience category = if('merge_final_table'[date hire reference]= 1 , if('merge_final_table'[absent] = 1,"undiscipline", if('merge_final_table'[Not Late (regular schedule)]=1,"discipline", if('merge_final_table'[Late (regular schedule)]= 1,"undiscipline", if('merge_final_table'[Not Late(regular schedule) with over time]= 1,"discipline", if('merge_final_table'[Late(regular schedule) with overtime]=1,"undiscipline", if('merge_final_table'[Not Late (regular schedule) but undertime]=1,"undiscipline", if('merge_final_table'[Late (regular schedule) and undertime]=1,"undiscipline", if('merge_final_table'[change schedule(early login with complete work schedule)]=1,"discipline", if('merge_final_table'[change schedule(early login with overtime)]=1,"discipline", if('merge_final_table'[change schedule(early login with undertime)]=1,"undiscipline", if('merge_final_table'[change schedule (late login regular logout)]=1,"undiscipline", if('merge_final_table'[change schedule (late login with overtime)]=1,"undiscipline", if('merge_final_table'[change schedule (late login and undertime)]=1,"undiscipline"))))))))))`
- Data Table:** A table with multiple columns. The column 'user obedience category' is highlighted in green. The data in this column consists of values like 'discipline' and 'undiscipline'.
- Right Panel:** The 'Data' pane on the right shows a list of columns, including 'user obedience category'.

In the Data View of merge_final_table, create a column for disciplined. Click New Column and type the DAX code,

`disciplined = if('merge_final_table'[user obedience category]="discipline",1)`

This column shows the disciplined user only.

The screenshot shows the Power BI Data View interface. The table 'merge_final_table' is displayed with columns: 'user obedience category', 'disciplined', 'age', 'sex', 'weight', 'height', and 'weight_kg'. The 'disciplined' column contains binary values (0 or 1) corresponding to the 'user obedience category'. The 'age' column contains values ranging from 18 to 25. The 'sex' column contains 'M' and 'F'. The 'weight' column contains values ranging from 50 to 100. The 'weight_kg' column contains values ranging from 10 to 20. The 'user obedience category' column contains values 'discipline' and 'undiscipline'.

In the Data View of merge_final_table, create a column for undisciplined. Click New Column and type the DAX code,

`undisciplined = if('merge_final_table'[user obedience category]="undiscipline",1)`

This column shows the undisciplined user only.

The screenshot shows the Power BI Data View interface. The table 'merge_final_table' is displayed with columns: 'user obedience category', 'undisciplined', 'age', 'sex', 'weight', 'height', and 'weight_kg'. The 'undisciplined' column contains binary values (0 or 1) corresponding to the 'user obedience category'. The 'age' column contains values ranging from 18 to 25. The 'sex' column contains 'M' and 'F'. The 'weight' column contains values ranging from 50 to 100. The 'weight_kg' column contains values ranging from 10 to 20. The 'user obedience category' column contains values 'discipline' and 'undiscipline'.

In the Data View of merge_final_table, create a column for annual. Click New Column and type the DAX code,

```
annual = if('merge_final_table'[leave_type] = "annual" &&  
    'merge_final_table'[date hire reference] = 1,1,0)
```

This column counts the number of annual leave of the user.

In the Data View of merge_final_table, create a column for day off. Click New Column and type the DAX code,

```
day off = if('merge_final_table'[leave_type] = "day off" &&  
    'merge_final_table'[date hire reference] = 1,1,0)
```

This column counts the number of day off leave of the user.

In the Data View of merge_final_table, create a column for compensatory. Click New Column and type the DAX code,

```
compensatory = if('merge_final_table'[leave_type] = "compensatory" &&  
    'merge_final_table'[date hire reference] = 1,1,0)
```

This column counts the number of compensatory leave of the user.

In the Data View of merge_final_table, create a column for sick. Click New Column and type the DAX code,

```
sick = if('merge_final_table'[leave_type] = "sick" &&  
    'merge_final_table'[date hire reference] = 1,1,0)
```

This column counts the number of sick leave of the user.

In the Data View of merge_final_table, create a column for special. Click New Column and type the DAX code,

```
special = if('merge_final_table'[leave_type] = "special" &&  
    'merge_final_table'[date hire reference] = 1,1,0)
```

This column counts the number of special leave of the user.

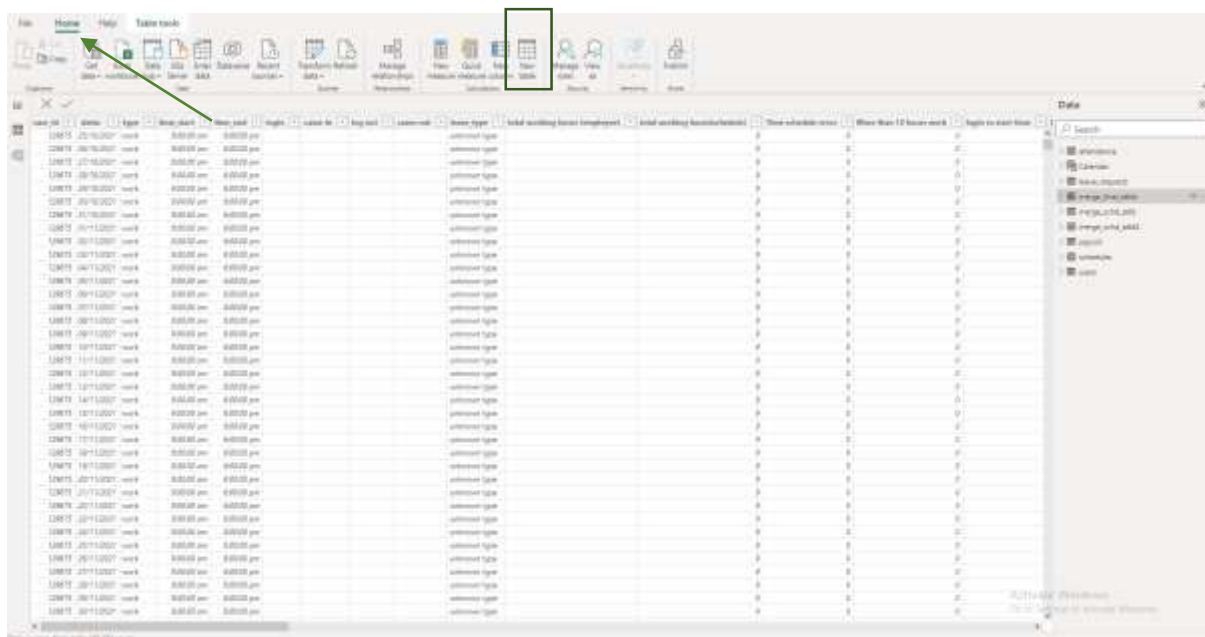
In the Data View of merge_final_table, create a column for unpaid. Click New Column and type the DAX code,

```
unpaid = if('merge_final_table'[leave_type] = "unpaid" &&  
    'merge_final_table'[date hire reference] = 1,1,0)
```

This column counts the number of unpaid leave of the user.

Creating New date table

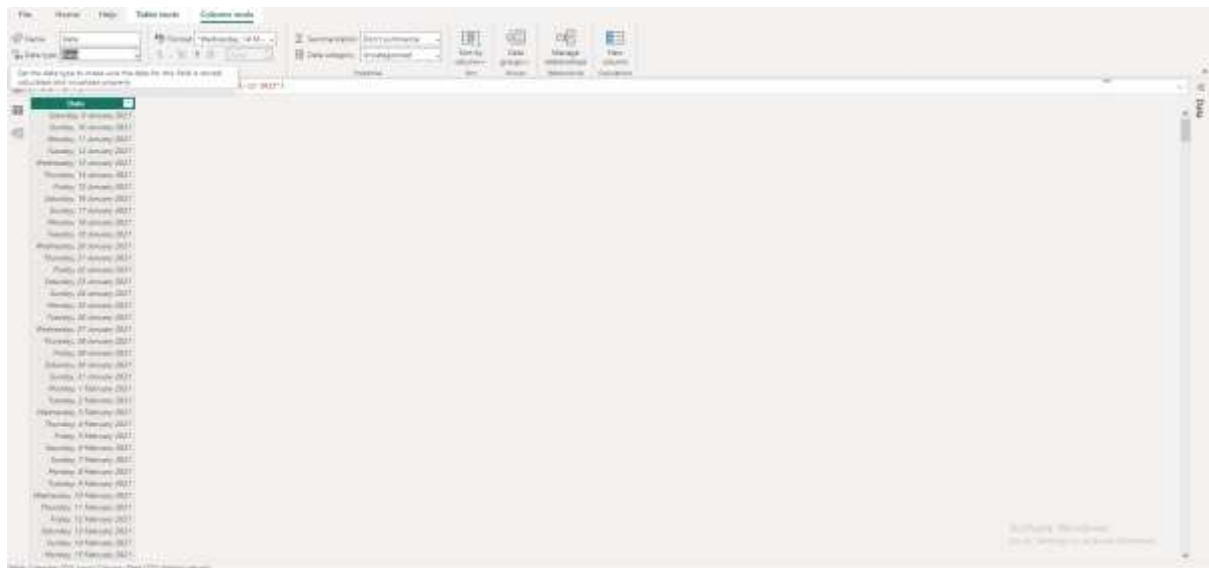
Go to Home and click New Table.



Input the DAX code,

```
Calendar = CALENDAR("01-09-2021","31-12-2022")
```

Change the data type to date.



Transform the Calendar table and create columns for Year, Month Number, Month, Weekday Number and Weekday.

Input DAX code,

Click new column and type

```
Year = YEAR('Calendar'[Date])
```

This column is for the Year

Click new column and type

```
Month = FORMAT('Calendar'[Date],"mmm")
```

This column is for the Month

Click new column and type

```
Weekday Number = WEEKDAY('Calendar'[Date],1)
```

This column is for the Weekday Number

Click new column and type

```
Weekday = FORMAT('Calendar'[Date],"ddd")
```

This column is for the Weekday

DAX Calculations

- absent count = CALCULATE(SUM('merge_final_table'[absent]))
- Late regular schedule count = CALCULATE(SUM ('merge_final_table'[Late (regular schedule)]))
- Late(regular schedule) and undertime count = CALCULATE(sum('merge_final_table'[Late (regular schedule) and undertime]))
- Late(regular schedule) with overtime count =
CALCULATE(SUM('merge_final_table'[Late(regular schedule) with overtime]))
- change schedule (late login regular logout) count =
CALCULATE(SUM('merge_final_table'[change schedule (late login regular logout)]))
- change schedule (late login and undertime) count =
CALCULATE(SUM('merge_final_table'[change schedule (late login and undertime)]))
- change schedule (late login with overtime) count =
CALCULATE(SUM('merge_final_table'[change schedule (late login with overtime)]))
- Not Late (regular schedule) count = CALCULATE(SUM('merge_final_table'[Not Late (regular schedule)]))
- Not Late(regular schedule) but undertime count = CALCULATE(SUM('merge_final_table'[Not Late (regular schedule) but undertime]))
- Not Late(regular schedule) with over time count = CALCULATE(SUM('merge_final_table'[Not Late(regular schedule) with over time]))
- change schedule(early login with regular logout) count =
CALCULATE(SUM('merge_final_table'[change schedule(early login with complete work schedule)]))
- change schedule(early login with undertime count =
CALCULATE(SUM('merge_final_table'[change schedule(early login with undertime)]))
- change schedule(early login with overtime) count =
CALCULATE(SUM('merge_final_table'[change schedule(early login with overtime)]))
- Total disciplined = CALCULATE(SUM('merge_final_table'[disciplined]))
- Total undisciplined = CALCULATE(SUM('merge_final_table'[undisciplined]))
- day off leave = CALCULATE(SUM([day off]))
- compensatory leave = CALCULATE(SUM([compensatory]))
- annual leave = CALCULATE(SUM([annual]))
- sick leave = CALCULATE(SUM([sick]))
- special leave = CALCULATE(SUM([special]))
- unpaid leave = CALCULATE(SUM([unpaid]))
- total leave = [annual leave]+[compensatory leave]+[day off leave]+[sick leave]+[special leave]+[unpaid leave]