1. Top Performer Identification by Department

- Task: Write a measure that returns the Employment_id of the top performer (highest Performance_Score) in each department.

```
Top Performer Employment ID =
VAR MaxScore =
    MAXX ( HR_Analytics, HR_Analytics[Performance_Score] )
RETURN
    MAXX (
        FILTER (
            HR_Analytics,
            HR_Analytics[Performance_Score] = MaxScore
        ),
        HR_Analytics[Employee_ID]
    )
```

Department	Top Performer Employment ID
Customer Support	99960
Engineering	99977
Finance	99942
HR	99999
IT	99997
Legal	99993
Marketing	99890
Operations	99865
Sales	99926
Total	**99999**

2. Year-over-Year Promotion Growth

- Task: Create a measure that calculates the % increase or decrease in promotions compared to the previous year.
- Assume Hire_Date is used as a reference for the year.

```
Promotion_rate_comparison =
var promoted = CALCULATE(COUNT(HR_Analytics[Employee_ID]),HR_Analytics[Promotions] = 1)
var promotion_rate = DIVIDE(promoted,[Number_of_employees])
var previous_year_rate = CALCULATE([Promotion rate],SAMEPERIODLASTYEAR('Date'[Date]))

RETURN DIVIDE([Promotion rate],previous_year_rate)
```

Year	Promotion_rate_comparison
2014	1.00
2015	1.00
2016	1.00
2017	1.00
2018	1.00
2019	1.00
2020	1.00
2021	1.00
2022	1.00
2023	1.00
2024	1.00
Total	**1.00**

3. Average Salary of Employees Who Resigned Within 2 Years

- Task: Calculate the average monthly salary of employees who resigned and had Years_at_company less than or equal to 2.

```
Last2_year_avg_salary =
var max_date = MAX(HR_Analytics[Hire_Date])
var avg_sal= CALCULATE(AVERAGE(HR_Analytics[Monthly_Salary]),DATESINPERIOD('Date'[Date],max_date,-2,YEAR))
RETURN avg_sal
```

4. Rank Employees by Satisfaction Score Within Their Department

- Task: Create a DAX measure or calculated column that ranks employees by Employee_Satisfaction_Score within their Department.

```
Satisfaction Rank_ =
RANKX (
    FILTER (
        HR_Analytics,
        HR_Analytics[Department] = EARLIER (HR_Analytics[Department])
    ),
    HR_Analytics[Employee_Satisfaction_Score],
    ,
    DESC,
    DENSE
)
```

Department	Employee_ID	Employee_Satisfaction_Score	Satisfaction Rank_
Customer Support	3373	5.00	1
Customer Support	6262	5.00	1
Customer Support	6654	5.00	1
Customer Support	8869	5.00	1
Customer Support	10286	5.00	1
Customer Support	18970	5.00	1
Customer Support	27056	5.00	1
Customer Support	35280	5.00	1
Customer Support	40005	5.00	1
Customer Support	41179	5.00	1

5. Correlation Between Training Hours and Performance

- Task: Calculate the Pearson correlation coefficient between Training_Hours and Performance_Score.

```
Correlation_Training_Performance =
DIVIDE (
    [Cov_Training_Performance],
    [Std_Training] * [Std_Performance],
    0
)
```

6. % of Employees Doing Remote Work Frequently

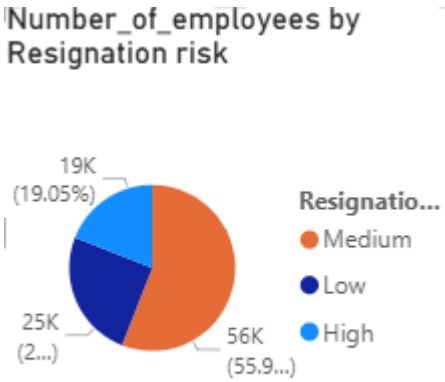
- Task: Write a measure that calculates the % of employees whose Remote_Work_Frequency is either "Weekly" or "Daily".

```
% of remote workers =
var remoters = CALCULATE(COUNT(HR_Analytics[Employee_ID]),FILTER(HR_Analytics,HR_Analytics[Remote_Work_Frequency] in {75,100}))
RETURN DIVIDE(remoters,[Number of employees])
```

9. Attrition Risk Index

- Task: Create a custom score to predict resignation risk using a formula like: `IF Employee_Satisfaction_Score < 3 && Overtime_Hours > 10 && Sick_Days > 5, "High", IF(Employee_Satisfaction_Score < 4, "Medium", "Low"`

```
Resignation risk = SWITCH(TRUE(),
                    HR_Analytics[Employee_Satisfaction_Score] < 3 && HR_Analytics[Overtime_Hours] > 10 && HR_Analytics[Sick_Days] > 5,"High",
                    HR_Analytics[Employee_Satisfaction_Score] < 4, "Medium", "Low")
```



10. Identify Overworked but Unpromoted Employees

- Task: Count employees who:
  - Have worked more than 45 Work_Hours_per_Week
  - Have more than 5 Overtime_Hours
  - Have Promotions = 0


```
Have more than 5 Overtime_Hours = CALCULATE(COUNT(HR_Analytics[Employee_ID]),HR_Analytics[Overtime_Hours] > 5)
```

```
Have_0_Promotions = CALCULATE(COUNT(HR_Analytics[Employee_ID]),HR_Analytics[Promotions] = 0)
```