Lesson 22

Topic: DAX Practice
Prerequisites: Download HR_Analytics.csv file

1. Top Performer Identification by Department

DAX:

DAX

CopyEdit

```
Top Performer ID =
CALCULATE (
    MAX (Employee_Performance[Employment_id]),
    TOPN (
        1,
        VALUES(Employee_Performance),
        Employee_Performance[Performance_Score],
        DESC
    )
)
```

Alternative using RANKX can be used if you want to display ranks.

---

2. Year-over-Year Promotion Growth

DAX:

DAX

CopyEdit

```
YoY Promotion Growth (%) =
VAR CurrentYear =
    CALCULATE(COUNT(Employee_Performance[Promotions]), YEAR(Employee_Performance[Hire_Date]) = YEAR(TODAY()))
VAR PreviousYear =
   CALCULATE(COUNT(Employee_Performance[Promotions]), YEAR(Employee_Performance[Hire_Date]) = YEAR(TODAY()) - 1)

RETURN

DIVIDE(CurrentYear - PreviousYear, PreviousYear, 0)
```

Assumes Hire_Date reflects year of promotion (simplified case).

---

3. Average Salary of Employees Who Resigned Within 2 Years

DAX:

DAX

CopyEdit

```
Avg Salary <2Y Resigned =

CALCULATE (

   AVERAGE(Employee_Performance[Monthly_Salary]),

   Employee_Performance[Resigned] = "Yes",

   Employee_Performance[Years_at_company] <= 2

)
```

---

4. Rank Employees by Satisfaction Score Within Their Department

DAX:

DAX

CopyEdit

```
Satisfaction Rank =

RANKX (

   FILTER (

      Employee_Performance,

      Employee_Performance[Department] = EARLIER(Employee_Performance[Department])

   ),

   Employee_Performance[Employee_Satisfaction_Score],
   ,
   DESC
)
```

---

5. Correlation Between Training Hours and Performance

Power BI doesn't have built-in Pearson correlation, but we can simulate it with DAX.

DAX (Pearson correlation):

DAX

CopyEdit

```
Correlation =
VAR MeanX = AVERAGE(Employee_Performance[Training_Hours])
VAR MeanY = AVERAGE(Employee_Performance[Performance_Score])
VAR Numerator =
    SUMX (
        Employee_Performance,
        (Employee_Performance[Training_Hours] - MeanX) *
        (Employee_Performance[Performance_Score] - MeanY)
    )
VAR Denominator =
    SQRT (
        SUMX (
            Employee_Performance,
            (Employee_Performance[Training_Hours] - MeanX) ^ 2
        ) *
        SUMX (
            Employee_Performance,
            (Employee_Performance[Performance_Score] - MeanY) ^ 2
        )
```

```
  )
RETURN
DIVIDE(Numerator, Denominator, 0)
```

---

6. % of Employees Doing Remote Work Frequently

DAX:

DAX

CopyEdit

```
% Remote Workers =
DIVIDE (
  CALCULATE (
    COUNTROWS(Employee_Performance),
    Employee_Performance[Remote_Work_Frequency] IN {"Weekly", "Daily"}
  ),
  COUNTROWS(Employee_Performance)
)
```

---

7. Employees With Consistently High Performance Over Tenure

DAX Concept:

If you simulate one score per year (e.g., divide Performance_Score / Years_at_company ≥ 4), you can write:

DAX

CopyEdit

```
Consistently High Performer =
VAR ScorePerYear =
  DIVIDE(Employee_Performance[Performance_Score], Employee_Performance[Years_at_company])
RETURN
IF(ScorePerYear >= 4, 1, 0)
```

Better if you have actual performance logs per year.

---

8. Department-Wise Salary Budget Utilization

Assume we have a related **Budget** table with fields: Department, Budget_Amount

DAX:

DAX

CopyEdit

```
Salary Utilization % =
VAR TotalSalary = SUM(Employee_Performance[Monthly_Salary])
VAR Budget = RELATED(Budget[Budget_Amount])
RETURN
DIVIDE(TotalSalary, Budget, 0)
```

---

9. Attrition Risk Index

DAX:

DAX

CopyEdit

```
Attrition Risk =
SWITCH(
   TRUE(),
   Employee_Performance[Employee_Satisfaction_Score] < 3 &&
   Employee_Performance[Overtime_Hours] > 10 &&
   Employee_Performance[Sick_Days] > 5, "High",

   Employee_Performance[Employee_Satisfaction_Score] < 4, "Medium",
   "Low"
)
```

---

10. Identify Overworked but Unpromoted Employees

DAX:

DAX

CopyEdit

```
Overworked No Promotion =
CALCULATE (
    COUNTROWS(Employee_Performance),
    Employee_Performance[Work_Hours_per_Week] > 45,
    Employee_Performance[Overtime_Hours] > 5,
    Employee_Performance[Promotions] = 0
)
```