1. Top Performer Identification by Department

Goal: Return the Employment_id of the top performer in each department.

```
Top Performer ID =
CALCULATE(
  FIRSTNONBLANK(Employee_Performance[Employment_id], 1),
  TOPN(
    1,
    Employee_Performance,
    Employee_Performance[Performance_Score],
    DESC
  ),
  ALLEXCEPT(Employee_Performance, Employee_Performance[Department])
)
```

- ALLEXCEPT keeps the department filter context, so it calculates the top performer **within each department**.

---

2. Year-over-Year Promotion Growth

Goal: % change in promotions vs previous year using Hire_Date.

```
Promotions YoY % =
VAR CurrentYearPromotions =
  CALCULATE(
    SUM(Employee_Performance[Promotions]),
    YEAR(Employee_Performance[Hire_Date]) = YEAR(TODAY())
  )
VAR PrevYearPromotions =
  CALCULATE(
    SUM(Employee_Performance[Promotions]),
    YEAR(Employee_Performance[Hire_Date]) = YEAR(TODAY()) - 1
  )
```

```
RETURN
DIVIDE(CurrentYearPromotions - PrevYearPromotions, PrevYearPromotions, 0)
```

- Format this as **Percentage** in Power BI.
- To make it dynamic with visuals, replace the YEAR() filter with SAMEPERIODLASTYEAR().

---

3. Average Salary of Employees Who Resigned Within 2 Years

```
Avg Salary Resigned <2yrs =
CALCULATE(
  AVERAGE(Employee_Performance[Monthly_Salary]),
  Employee_Performance[Resigned] = "Yes",
  Employee_Performance[Years_at_company] <= 2
)
```

- Returns the **average monthly salary** for employees who resigned with ≤2 years of tenure.

---

4. Rank Employees by Satisfaction Score Within Department

Use a **calculated column** for stable ranking:

```
Rank by Satisfaction =
RANKX(
  FILTER(
    Employee_Performance,
    Employee_Performance[Department] =
EARLIER(Employee_Performance[Department])
  ),
  Employee_Performance[Employee_Satisfaction_Score],
  ,
  DESC,
  DENSE
)
```

- DENSE ensures no rank gaps if two employees have the same score.

---

5. Correlation Between Training Hours and Performance

Pearson correlation:

```
Correlation Training vs Performance =
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)
```

- Result: **-1 to +1** correlation coefficient.

---

6. % of Employees Doing Remote Work Frequently

```
% Remote Work Frequently =
```

```
DIVIDE(
  CALCULATE(
    COUNTROWS(Employee_Performance),
    Employee_Performance[Remote_Work_Frequency] IN {"Weekly", "Daily"}
  ),
  COUNTROWS(Employee_Performance),
  0
)
```

- Returns % of employees who work remotely **weekly or daily**.

---

7. Employees With Consistently High Performance Over Tenure

Assume tenure approximates years and we check if the **Performance_Score ≥ 4** each year:

```
Consistently High Performers =
CALCULATE(
  COUNTROWS(Employee_Performance),
  FILTER(
    Employee_Performance,
    Employee_Performance[Performance_Score] >= 4 &&
    Employee_Performance[Years_at_company] >= 1
  )
)
```

- If you have historical scores by year, use **ALLSELECTED** by year for strict checks.

---

8. Department-Wise Salary Budget Utilization

Assume you have a DepartmentBudget table with [Department] and [Budget]:

```
Salary Budget Utilization =
DIVIDE(
  SUM(Employee_Performance[Monthly_Salary]),
  RELATED(DepartmentBudget[Budget]),
```

```
    0
)
```

- Gives % of budget used per department.

---

9. Attrition Risk Index

Use a **calculated column** for risk classification:

```
Attrition Risk =
IF(
    Employee_Performance[Employee_Satisfaction_Score] < 3 &&
    Employee_Performance[Overtime_Hours] > 10 &&
    Employee_Performance[Sick_Days] > 5,
    "High",
    IF(Employee_Performance[Employee_Satisfaction_Score] < 4, "Medium", "Low")
)
```

- Returns "High", "Medium", or "Low" per employee.

---

10. Identify Overworked but Unpromoted Employees

```
Overworked Unpromoted Count =
CALCULATE(
    COUNTROWS(Employee_Performance),
    Employee_Performance[Work_Hours_per_Week] > 45,
    Employee_Performance[Overtime_Hours] > 5,
    Employee_Performance[Promotions] = 0
)
```

- Count of employees meeting all three conditions.