DAX Tasks and Questions – Solutions

1. Use variable to calculate % Growth in Sales Compared to Last Year

```
YoY Growth % =
VAR CurrentSales = SUM('Chocolate Sales (1)'[Amount])
VAR LastYearSales =
  CALCULATE(
    SUM('Chocolate Sales (1)'[Amount]),
    SAMEPERIODLASTYEAR('Calendar'[Date])
  )
RETURN
DIVIDE(CurrentSales - LastYearSales, LastYearSales)
```

2. Use variable to calculate the difference between Sales Amount of current month and previous month

```
Month Sales Difference =
VAR CurrentMonthSales =
  CALCULATE(
    SUM('Chocolate Sales (1)'[Amount]),
    MONTH('Calendar'[Date]) = MONTH(MAX('Calendar'[Date])) &&
    YEAR('Calendar'[Date]) = YEAR(MAX('Calendar'[Date]))
  )
VAR PreviousMonthSales =
  CALCULATE(
    SUM('Chocolate Sales (1)'[Amount]),
    PARALLELPERIOD('Calendar'[Date], -1, MONTH)
  )
RETURN
CurrentMonthSales - PreviousMonthSales
```

3. Calculate total boxes shipped and average monthly boxes in one measure using VAR

```
Total and Avg Boxes =
VAR TotalBoxes = SUM('Chocolate Sales (1)'[Boxes Shipped])
VAR MonthCount = DISTINCTCOUNT('Calendar'[Month])
VAR AvgBoxes = ROUND(DIVIDE(TotalBoxes, MonthCount), 0)
RETURN
"Total: " & TotalBoxes & UNICHAR(10) & "Avg/Month: " & AvgBoxes
```

4. Calculate total boxes shipped and average monthly boxes in one measure using VAR and return average monthly boxes.

```
Avg Monthly Boxes =
VAR TotalBoxes = SUM('Chocolate Sales (1)'[Boxes Shipped])
VAR MonthCount = DISTINCTCOUNT('Calendar'[Month])
RETURN ROUND(DIVIDE(TotalBoxes, MonthCount), 0)
```

5. Calculate growth percentage from last month.

```
Last Month Growth % =
VAR CurrentMonthSales =
  CALCULATE(
    SUM('Chocolate Sales (1)'[Amount]),
    MONTH('Calendar'[Date]) = MONTH(MAX('Calendar'[Date])) &&
    YEAR('Calendar'[Date]) = YEAR(MAX('Calendar'[Date]))
  )
VAR PreviousMonthSales =
  CALCULATE(
    SUM('Chocolate Sales (1)'[Amount]),
    PARALLELPERIOD('Calendar'[Date], -1, MONTH)
  )
RETURN
DIVIDE(CurrentMonthSales - PreviousMonthSales, PreviousMonthSales)
```

6. Create a moving average of sales over the last 3 months.

```
Moving Avg 3 Months =
VAR SalesLast3Months =
  CALCULATE(
    SUM('Chocolate Sales (1)'[Amount]),
    DATESINPERIOD(
      'Calendar'[Date],
      MAX('Calendar'[Date]),
      -3,
      MONTH
    )
  )
RETURN DIVIDE(SalesLast3Months, 3)
```

7. Use Card to show a Dynamic Message Based on Sales Rank and YoY Performance.

```
Performance Message =
VAR SelectedProduct = SELECTEDVALUE('Chocolate Sales (1)'[Product])
VAR CurrentSales =
  CALCULATE(
    SUM('Chocolate Sales (1)'[Amount]),
    'Chocolate Sales (1)'[Product] = SelectedProduct
```

```
  )
VAR LastYearSales =
  CALCULATE(
    SUM('Chocolate Sales (1)'[Amount]),
    SAMEPERIODLASTYEAR('Calendar'[Date]),
    'Chocolate Sales (1)'[Product] = SelectedProduct
  )
VAR YoYGrowth = DIVIDE(CurrentSales - LastYearSales, LastYearSales)
VAR ProductRank =
  RANKX(
    ALL('Chocolate Sales (1)'[Product]),
    CALCULATE(SUM('Chocolate Sales (1)'[Amount]))
  )
RETURN
SWITCH(
  TRUE(),
  ProductRank = 1 && YoYGrowth > 0,
    "Top Performer – Sales up by " & FORMAT(YoYGrowth, "0%"),
  YoYGrowth > 0,
    "Consistent Performer",
  TRUE,
    "Needs Improvement"
)
```

8. List Top 5 tips to optimize DAX query manually and explain why you choose.

1. Use variables (VAR) to store intermediate results → Improves readability and performance.
2. Minimize row context with CALCULATE → Prevents unnecessary row-by-row evaluation.
3. Avoid using FILTER unnecessarily → Use direct column filters when possible.
4. Prefer SUMX over FILTER+SUM if you can write a simple expression.
5. Reduce cardinality of columns → Less memory, faster performance.

9. What is the benefit of using DAX optimization tools like DAX Studio, Performance Analyzer, Tabular Editor

- DAX Studio → Shows exact query plan and timings; helps identify bottlenecks.
- Performance Analyzer → Built into Power BI; helps track expensive visuals/measures.
- Tabular Editor → Lightweight and efficient for managing measures, formatting DAX, and batch editing.

10. Create a flag (Yes/No) if a product is in the top 5 by total sales. Use RANKX in a variable; avoid calculating rank more than once.

```
Top 5 Flag =
VAR ProductRank =
   RANKX(
      ALL('Chocolate Sales (1)'[Product]),
      CALCULATE(SUM('Chocolate Sales (1)'[Amount]))
   )
RETURN IF(ProductRank <= 5, "Yes", "No")
```