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

You can use the VAR keyword in DAX to store intermediate results and avoid recalculations. For example:

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

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

This measure calculates the difference between the current month's sales and the previous month's sales using VAR:

```
Sales Diff MoM =
VAR CurrentMonthSales = CALCULATE(SUM('Chocolate Sales (1)'[Amount]),
MONTH('Date'[Date]) = MONTH(TODAY()), YEAR('Date'[Date]) = YEAR(TODAY()))
VAR PrevMonthSales = CALCULATE(SUM('Chocolate Sales (1)'[Amount]),
MONTH('Date'[Date]) = MONTH(EDATE(TODAY(), -1)), YEAR('Date'[Date]) =
YEAR(EDATE(TODAY(), -1)))
RETURN CurrentMonthSales - PrevMonthSales
```

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

```
Total & Avg Monthly Boxes =
VAR TotalBoxes = SUM('Chocolate Sales (1)'[Boxes Shipped])
VAR MonthsCount = DISTINCTCOUNT('Date'[Month])
VAR AvgMonthlyBoxes = DIVIDE(TotalBoxes, MonthsCount)
RETURN "Total: " & FORMAT(TotalBoxes, "#,##0") & " | Avg: " & FORMAT(AvgMonthlyBoxes,
"#,##0")
```

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 MonthsCount = DISTINCTCOUNT('Date'[Month])
RETURN DIVIDE(TotalBoxes, MonthsCount)
```

5. Calculate growth percentage from last month

```
Growth % MoM =
VAR CurrentMonthSales = CALCULATE(SUM('Chocolate Sales (1)'[Amount]),
MONTH('Date'[Date]) = MONTH(TODAY()), YEAR('Date'[Date]) = YEAR(TODAY()))
VAR PrevMonthSales = CALCULATE(SUM('Chocolate Sales (1)'[Amount]),
MONTH('Date'[Date]) = MONTH(EDATE(TODAY(), -1)), YEAR('Date'[Date]) =
YEAR(EDATE(TODAY(), -1)))
```

```
RETURN DIVIDE(CurrentMonthSales - PrevMonthSales, PrevMonthSales)
```

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

```
Sales 3M Moving Avg =
VAR Last3Months = DATESINPERIOD('Date'[Date], MAX('Date'[Date]), -3, MONTH)
RETURN CALCULATE(AVERAGEX(VALUES('Date'[Month]), CALCULATE(SUM('Chocolate Sales (1)'[Amount]))), Last3Months)
```

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

```
Performance Message =
VAR SalesCurrentYear = CALCULATE(SUM('Chocolate Sales (1)'[Amount]), YEAR('Date'[Date]) = YEAR(TODAY()))
VAR SalesLastYear = CALCULATE(SUM('Chocolate Sales (1)'[Amount]), YEAR('Date'[Date]) = YEAR(TODAY()) - 1)
VAR GrowthPct = DIVIDE(SalesCurrentYear - SalesLastYear, SalesLastYear)
VAR RankProduct = RANKX(ALL('Chocolate Sales (1)'[Product]), CALCULATE(SUM('Chocolate Sales (1)'[Amount])), , DESC)
RETURN IF(RankProduct <= 5 && GrowthPct > 0.1,
"Top Performer - Sales up by " & FORMAT(GrowthPct, "0%"),
IF(GrowthPct >= 0, "Consistent Performer", "Needs Improvement"))
```

8. List Top 5 tips to optimize DAX query manually

1. Use VAR to store intermediate results.
2. Remove unused columns to reduce model size.
3. Filter on the smallest necessary dataset.
4. Use ALL and REMOVEFILTERS only when required.
5. Create supporting tables for complex calculations.

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

These tools help you monitor, analyze, and improve DAX performance:
- **DAX Studio**: Analyze query plans, measure execution time, and identify bottlenecks.
- **Performance Analyzer**: Measure visual performance inside Power BI.
- **Tabular Editor**: Manage and optimize your data model, write and test DAX quickly.

10. Create a flag (Yes/No) if a product is in the top 5 by total sales

```
Top5 Flag =
VAR ProductRank = RANKX(ALL('Chocolate Sales (1)'[Product]), CALCULATE(SUM('Chocolate Sales (1)'[Amount])), , DESC)
RETURN IF(ProductRank <= 5, "Yes", "No")
```