Basic / Intermediate / Advanced (1–10)

1. % Growth in Sales Compared to Last Year (with VAR)

```
Sales Growth % YoY =
VAR CurrentSales = [Total Sales]
VAR LastYearSales =
    CALCULATE ( [Total Sales], SAMEPERIODLASTYEAR ( 'Date'[Date] ) )
RETURN
DIVIDE ( CurrentSales - LastYearSales, LastYearSales )
```

---

2. Difference Between Sales Amount of Current Month vs Previous Month (with VAR)

```
Sales Diff MoM =
VAR CurrentMonthSales = CALCULATE ( [Total Sales], DATESMTD ( 'Date'[Date] )
)
VAR PrevMonthSales =
    CALCULATE ( [Total Sales], PARALLELPERIOD ( 'Date'[Date], -1, MONTH ) )
RETURN
CurrentMonthSales - PrevMonthSales
```

---

3. Total Boxes and Average Monthly Boxes (both in one measure, using VAR)

```
Boxes Summary =
VAR TotalBoxes = SUM ( ChocolateSales[Boxes] )
VAR AvgMonthlyBoxes =
    AVERAGEX (
        VALUES ( 'Date'[MonthYear] ),
        CALCULATE ( SUM ( ChocolateSales[Boxes] ) )
    )
RETURN
"Total: " & FORMAT ( TotalBoxes, "#,##0" ) &
" | Avg/Month: " & FORMAT ( AvgMonthlyBoxes, "#,##0" )
```

---

4. Only Return Average Monthly Boxes (using VAR)

```
Avg Monthly Boxes =
VAR TotalBoxes = SUM ( ChocolateSales[Boxes] )
VAR AvgMonthlyBoxes =
    AVERAGEX (
        VALUES ( 'Date'[MonthYear] ),
        CALCULATE ( SUM ( ChocolateSales[Boxes] ) )
    )
RETURN
AvgMonthlyBoxes
```

---

5. Growth % from Last Month

```
Sales Growth % MoM =
VAR CurrentMonthSales = CALCULATE ( [Total Sales], DATESMTD ( 'Date'[Date] )
)
VAR PrevMonthSales =
    CALCULATE ( [Total Sales], PARALLELPERIOD ( 'Date'[Date], -1, MONTH ) )
RETURN
DIVIDE ( CurrentMonthSales - PrevMonthSales, PrevMonthSales )
```

---

6. Moving Average of Sales (Last 3 Months)

```
Sales 3M Moving Avg =
AVERAGEX (
    DATESINPERIOD ( 'Date'[Date], MAX ( 'Date'[Date] ), -3, MONTH ),
    [Total Sales]
)
```

---

7. Dynamic Message Based on Sales Rank + YoY Performance

```
Performance Message =
VAR Product = SELECTEDVALUE ( ChocolateSales[Product] )
VAR RankProduct =
    RANKX ( ALL ( ChocolateSales[Product] ), [Total Sales], , DESC )
VAR SalesCY = [Total Sales]
VAR SalesLY =
    CALCULATE ( [Total Sales], SAMEPERIODLASTYEAR ( 'Date'[Date] ) )
VAR YoYGrowth = DIVIDE ( SalesCY - SalesLY, SalesLY )
RETURN
SWITCH (
    TRUE(),
    RankProduct <= 3 && YoYGrowth > 0, "Top Performer - Sales up by " &
FORMAT ( YoYGrowth, "0%" ),
    YoYGrowth >= 0, "Consistent Performer",
    "Needs Improvement"
)
```

---

8. Top 5 Tips to Optimize DAX Queries (manual optimization)

1. **Use variables (`VAR`)** → Avoid repeating the same calculation multiple times (better readability + performance).
2. **Reduce filter context inside CALCULATE** → Use `KEEPFILTERS` or limit `FILTER` to relevant columns.
3. **Prefer iterators (X functions) only when necessary** → Use `SUMX` or `AVERAGEX` only when row context is needed.
4. **Leverage pre-aggregations in Power Query** → Push transformations upstream to reduce calculation load in DAX.
5. **Avoid using ALL unnecessarily** → Restoring the entire table context can be expensive. Use `REMOVEFILTERS` more selectively.

👉 I picked these because they directly reduce query engine load, avoid repeated scans, and make formulas both faster and cleaner.

---

9. Benefits of DAX Optimization Tools

- **DAX Studio** → Analyze query plans, see performance bottlenecks, measure query duration.
- **Performance Analyzer (Power BI)** → Shows which visuals/measures are slow, so you target optimization.

- **Tabular Editor** → Manage measures, calculation groups, and apply best practices quickly.

☑ Together, they help you identify bottlenecks and apply systematic optimizations.

---

10. Flag (Yes/No) if Product is in Top 5 by Sales (using VAR & RANKX once)

```
Top 5 Product Flag =
VAR ProductRank =
    RANKX ( ALL ( ChocolateSales[Product] ), [Total Sales], , DESC )
RETURN
IF ( ProductRank <= 5, "Yes", "No" )
```