

Power BI DAX Interview Questions - CodeInQueries

1. Calculate the number of days between a customer's first and last purchase.

```
DATEDIFF(
    CALCULATE(MIN(Sales[Date])),
    Sales[CustomerID]),
    CALCULATE(MAX(Sales[Date])),
    Sales[CustomerID]),
    DAY
)
```

2. Find the top 2 best-selling products per category dynamically.

```
TOPN(2,
    FILTER(
        SUMMARIZE(ALL(Sales), Sales[Category],
            Sales[ProductID], "TotalSales", SUM(Sales[SalesAmount])),
        Sales[Category] = MAX(Sales[Category])
    ),
    [TotalSales],
    DESC
)
```

3. Count active customers who made at least one purchase in the last 12 months.

```
CALCULATE(
    DISTINCTCOUNT(Sales[CustomerID]),
    FILTER(ALL(Sales), Sales[Date] >= TODAY() - 365)
)
```

4. Calculate the total revenue contributed by the top 20% of transactions.

```
CALCULATE(
    SUM(Sales[SalesAmount]),
    FILTER(
        ALL(Sales),
        RANKX(ALL(Sales), Sales[SalesAmount], , DESC) <=
COUNTROWS(Sales) * 0.2
    )
)
```

5. Calculate the average time gap (in days) between consecutive purchases for each customer.

```
AVERAGEX(
    FILTER(
        ADDCOLUMNS(
            Sales,
            "PreviousPurchaseDate",
            CALCULATE(MAX(Sales[Date]), Sales[Date] < EARLIER(Sales[Date]))
        ),
        NOT(ISBLANK([PreviousPurchaseDate]))
    ),
    DATEDIFF([PreviousPurchaseDate], Sales[Date], DAY)
)
```

6. Identify customers who purchased in at least 2 different product categories.

```
CALCULATE(
    DISTINCTCOUNT(Sales[CustomerID]),
    FILTER(
        VALUES(Sales[CustomerID]),
        CALCULATE(DISTINCTCOUNT(Sales[Category])) >= 2
    )
)
```

7. Find the most recent sales amount before a selected date.

```
CALCULATE(
    SUM(Sales[SalesAmount]),
    FILTER(ALL(Sales), Sales[Date] <
SELECTEDVALUE(Sales[Date]))
)
```

8. Calculate the cumulative distinct count of customers over time.

```
CALCULATE(
    DISTINCTCOUNT(Sales[CustomerID]),
    FILTER(ALL(Sales), Sales[Date] <= MAX(Sales[Date]))
)
```

9. Get the last non-empty sales value for each product.

```
LOOKUPVALUE(
    Sales[SalesAmount],
    Sales[Date],
    MAXX(FILTER(ALL(Sales),
    NOT(ISBLANK(Sales[SalesAmount])) && Sales[ProductID] =
```

```
MAX(Sales[ProductID]), Sales[Date])
)
```

10. Determine the sales percentage change between the first and last months of each year.

```
(DIVIDE (
    CALCULATE (SUM(Sales[SalesAmount]),
    ENDOFYEAR(Sales[Date])),
    CALCULATE (SUM(Sales[SalesAmount]),
    STARTO FYEAR(Sales[Date]))
) - 1) * 100
)
```

11. Calculate the year-over-year growth percentage for total sales.

```
YoY Growth =
VAR CurrentYearSales = SUM(Sales[SalesAmount])
VAR PreviousYearSales = CALCULATE (SUM(Sales[SalesAmount]),
SAMEPERIODLASTYEAR (Sales[Date]))
RETURN
DIVIDE (CurrentYearSales - PreviousYearSales,
PreviousYearSales, 0) * 100
```

12. Compute the running total of sales.

```
RunningTotal =
CALCULATE (
    SUM(Sales[SalesAmount]),
    FILTER(ALL (Sales), Sales[Date] <= MAX (Sales[Date]))
)
```

13. Find the average sales per day over the last 30 days.

```
AvgSalesLast30Days =
AVERAGEX (
    DATESINPERIOD (Sales[Date], TODAY(), -30, DAY),
    CALCULATE (SUM(Sales[SalesAmount]))
)
```

14. Get the last transaction date for each customer.

```
LastPurchaseDate =
CALCULATE (MAX (Sales[Date]), ALLEXCEPT (Sales,
Sales[CustomerID]))
```

15. Rank products by sales within each category.

```

ProductRank =
    RANKX (
        FILTER (ALL (Sales) , Sales[Category] =
MAX (Sales[Category])) ,
        SUM (Sales[SalesAmount]) , , DESC , DENSE
    )

```

16. Find the cumulative sales percentage by product.

```

CumulativeSalesPercentage =
    VAR TotalSales = CALCULATE (SUM (Sales[SalesAmount]) ,
ALL (Sales))
    VAR ProductSales = SUM (Sales[SalesAmount])
    RETURN
    DIVIDE (ProductSales, TotalSales, 0) * 100

```

17. Calculate the retention rate of customers month over month.

```

RetentionRate =
    VAR CurrentMonthCustomers = DISTINCTCOUNT (Sales[CustomerID])
    VAR PreviousMonthCustomers =
        CALCULATE (DISTINCTCOUNT (Sales[CustomerID]) ,
PARALLELPERIOD (Sales[Date] , -1 , MONTH))
    RETURN
    DIVIDE (CurrentMonthCustomers, PreviousMonthCustomers, 0)

```

18. Find the total number of customers who made purchases in consecutive months.

```

RepeatedCustomers =
    CALCULATE (
        DISTINCTCOUNT (Sales[CustomerID]) ,
        FILTER (ALL (Sales) ,
            Sales[Date] >= EOMONTH (MAX (Sales[Date]) , -1)
            && Sales[Date] <= MAX (Sales[Date])
        )
    )

```

19. Find customers who made at least 3 purchases in the last 6 months.

```

FrequentBuyers =
    CALCULATE (
        DISTINCTCOUNT (Sales[CustomerID]) ,
        FILTER (
            VALUES (Sales[CustomerID]) ,
            CALCULATE (COUNT (Sales[Date]) ,
                FILTER (ALL (Sales) , Sales[Date] >= TODAY () - 180)
            ) >= 3
        )
    )

```


20. Calculate the sales contribution of each customer to the total revenue.

```
SalesContribution =
    VAR CustomerSales = SUM(Sales[SalesAmount])
    VAR TotalSales = CALCULATE(SUM(Sales[SalesAmount]),
ALL(Sales))
    RETURN
    DIVIDE(CustomerSales, TotalSales, 0) * 100
```

21. Find the total sales for the latest available date.

```
LatestSales
VAR MaxDate = MAX(Sales[Date])
RETURN
CALCULATE(SUM(Sales[SalesAmount]), Sales[Date] = MaxDate)
```

22. Find the total revenue generated in the first 3 months of each year.

```
Q1 Sales =
    CALCULATE (
        SUM(Sales[SalesAmount]),
        FILTER(ALL(Sales), MONTH(Sales[Date]) <= 3)
    )
```

23. Find the last 3 months' rolling sales.

```
Rolling3MonthsSales =  
    CALCULATE(  
        SUM(Sales[SalesAmount]),  
        DATESINPERIOD(Sales[Date], MAX(Sales[Date]), -3, MONTH)  
    )
```

24. Calculate the stock level at any given date.

```

StockLevel =
    VAR TotalPurchases =
        CALCULATE(SUM(Inventory[PurchaseQuantity]), ALL(Inventory))
    VAR TotalSales = CALCULATE(SUM(Sales[SalesQuantity]), ALL(Sales))
    RETURN
        TotalPurchases - TotalSales

```

25. Find the first date when stock levels dropped below a threshold (e.g., 100 units).

```
FirstLowStockDate =  
    CALCULATE (  
        MIN(Inventory[Date]),  
        FILTER(ALL(Inventory), Inventory[StockLevel] < 100)  
    )
```

**Follow @codeinquiries
for more interview
questions**