```
TotalSales      SUMX ( Ordertbl, Ordertbl[Amount] * Ordertbl[SellingPrice] )
TotalCost       SUMX ( Ordertbl, Ordertbl[Amount] * Ordertbl[PurchasingPrice] )
TotalProfit     Measurestbl[TotalSales] - Measurestbl[TotalCost]
ProfitMargin    DIVIDE(Measurestbl[TotalProfit],Measurestbl[TotalSales])
ParallelPeriodSales     CALCULATE (
           Measurestbl[TotalSales],
           PARALLELPERIOD ( Datestbl[Date], -1, QUARTER )
        )
Ranking IF (
           ISINSCOPE ( Ordertbl[Category] ),
           RANKX ( ALL ( Ordertbl[Category] ), Measurestbl[TotalSales],, DESC,
DENSE ),
           BLANK ()
        )
TopXSales       IF (
           Measurestbl[Ranking] <= RankingParameter[RankingParameter Value],
           Measurestbl[TotalSales],
           BLANK ()
        )
CumSales        CALCULATE (
           Measurestbl[TotalSales],
           FILTER (
               ALLSELECTED ( Datestbl[Date] ),
               Datestbl[Date] <= MAX ( Ordertbl[OrderDate] )
           )
        )
TotalSalesLY
CALCULATE(Measurestbl[TotalSales],DATEADD(Datestbl[Date],-1,YEAR))
CumSalesLY      CALCULATE (
           Measurestbl[TotalSalesLY],
           FILTER (
               ALLSELECTED ( Datestbl[Date] ),
               Datestbl[Date] <= MAX ( Datestbl[Date] )
           )
        )
TotalScenarioSales
SUMX(Ordertbl,Ordertbl[Amount]*(1+DemandChange[DemandChange
Value])*Ordertbl[SellingPrice]*(1+PriceChange[PriceChange Value]))
Rolling30DaySales       CALCULATE (
           Measurestbl[TotalSales],
           FILTER (
               ALLSELECTED ( Datestbl[Date] ),
               Datestbl[Date]
                   > MAX ( Datestbl[Date] ) - 30
                   && Datestbl[Date] <= MAX ( Datestbl[Date] )
           )
        )
Top10Clients    CALCULATE (
           Measurestbl[TotalProfit],
           FILTER (
               VALUES ( Ordertbl[CustomerCompany] ),
               RANKX (
                   ALL ( Ordertbl[CustomerCompany] ),
                   Measurestbl[TotalProfit],
```

```
                    ,
                    DESC,
                    DENSE
                ) <= 10
            )
        )
UniqueCustomers DISTINCTCOUNT(Ordertbl[CustomerCompany])
ClassifyCustomers       CALCULATE (
            Measurestbl[TotalSales],
            FILTER (
                VALUES ( Ordertbl[CustomerCompany] ),
                COUNTROWS (
                    FILTER (
                        CustomerClassificationtbl,
                        RANKX ( ALL ( Ordertbl[CustomerCompany] ),
Measurestbl[TotalSales],, DESC ) >= CustomerClassificationtbl[Min]
                            && RANKX ( ALL ( Ordertbl[CustomerCompany] ),
Measurestbl[TotalSales],, DESC ) <= CustomerClassificationtbl[Max]
                    )
                ) > 0
            )
        )
TotalSalesOnWeekends    VAR Weekends =
            FILTER ( Datestbl, OR ( Datestbl[DayOfWeek] = 5, Datestbl[DayOfWeek]
= 6 ) )
        RETURN
            SUMX ( Weekends, Measurestbl[TotalSales] )
TotalSalesOnWeekdays    VAR Weekdays =
            FILTER ( Datestbl, AND ( Datestbl[DayOfWeek] <> 5,
Datestbl[DayOfWeek] <> 6 ) )
        RETURN
            SUMX ( Weekdays, Measurestbl[TotalSales] )
RankingParameter Value  SELECTEDVALUE('RankingParameter'[RankingParameter])
PriceChange Value       SELECTEDVALUE('PriceChange'[PriceChange], 0)
DemandChange Value      SELECTEDVALUE('DemandChange'[DemandChange], 0)
```