Configuración general

CONFIG.HIERARCHY.INCLUDE_MEMBERS_NOT_IN_HIERARCHY

Se usa cuando se quiere que los cálculos posteriores también se realicen para miembros de la dimensión que no están en la jerarquía

```
CONFIG.HI ERA RCH Y.I NCL UDE _ME MBE RS_ NOT _IN _HI ERARCHY = [d/t.S :SA P_A LL_ TRA DIN GPA RTNER]
```

CONFIG.GENERATE_UNBOOKED_DATA

Sirve para indicar si se tienen en cuenta o no los datos vacios. Se configura con ON/OFF, de manera predeterminada estará en OFF

```
CONFIG.GE NER ATE _UN BOO KED _DATA = ON
```

Selección de miembros

MEMBERSET

Se establecen qué valores de las dimensiones/métricas se utilizarán o excluirán en cálculos posteriores de manera general

```
MEMBERSET [d/AUD IT_ TRAIL] ="SA C"
```

En el siguiente ejemplo se excluyen de la dimensión Producto los hijos de la jerarquía = All_Clothes

```
MEMBERSET [d/PRO DUCT] != BASEME MBER( [d/PRO DUC T].[ h/H1], " All _Cl oth es")
```

BASEMEMBER

Se utiliza en la segunda parte de MEMBERSET cuando se quiere indicar el padre en específico de una jerarquía

```
MEMBERSET [d/SAP _FI _S4 HC_ GLA CCOUNT] = (BASEM EMB ER( [d/ ACC OUN T].[ h/ Cla se_ coste], " Ing res os
"))
```

Agregación de miembros

AGGREGATE_DIMENSIONS

Sirve para definir qué dimensiones se van a agregar antes de realizar el cálculo

```
AGGREG ATE _DI MEN SIONS = [d/t.S :SA P_A LL_ COS TCE NTER]
```

AGGREGATE_WRITETO

Se indica qué valor de la dimensión agregada le asignamos a cada uno de los miembros

```
AGGREG ATE _WR ITETO  [d/t.S :SA P_A LL_ COS TCE NTER] = " #"
```


Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish Yours!
https://apollopad.com

Escribir datos

RESULTLOOKUP

Devuelve los valores para los miembros establecidos previamente

```
RESULT LOO KUP()
```

También se pueden indicar valores concretos dentro del propio RESULTLOOKUP

```
RESULT LOO KUP ([d /Ve rsion] = " pub lic.Ac tua l")
```

DATA

Se usa para sobrescribir y guardar datos. Se pueden hacer cálculos con los RESULTLOOKUP y guardarlos en la cuenta que se quiera

```
DATA([ d/A ccount] = " Tas a")) = RESULT LOO KUP ([d /Ac count] = " Cuenta 1") / RESULT LOO KUP ([d /Ac coun
```

Dentro de DATA se pueden indicar en qué valores de la variable se van a guardar los datos

```
DATA([ d/A ccount] = " Tas a",[ d/T ime ]="2 023 04", [d/SAP _AL L_C OUN TRY DEL IVERY] ="#", [d/ SAP _CO NT
 ([d /Ve rsion] = " pub lic.Ac tua l")
```

DELETE

Permite borrar datos

```
DELETE ([d /SA P_F I_S 4HC _GL ACC OUNT] = " Tas a")
```

También se pueden indicar valores concretos si no se ha hecho previamente

```
DELETE ([d /SA P_F I_S 4HC _GL ACC OUNT] = " Tas a", [d/SAP _AL L_C OUN TRY DEL IVERY] ="#")
```

Selección de tiempo/periodo

PREVIOUS

Selecciona un tiempo concreto o un periodo de la dimensión fecha.

```
DATA() = RESULT LOO KUP ([d /Date] = PREVIO US(1, " MON TH", " 202 0-0 4-2 1"))
```

El primer argumento nos indica cuántas unidades, el segundo la granularidad y el tercero que fecha tomamos como referencia. Por tanto, nos devolvería como fecha el mes previo a abril: mayo.

Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish Yours!
https://apollopad.com

Bucles

FOREACH

Repite la sentencia para cada miembro de la dimensión que se le indique. Con ENDFOR se cierra el bucle

```
FOREACH [d/Date]
   DATA() = RESULT LOO KUP ([d /DATE] = PREVIO US(1)) * 1.1
 ENDFOR
```

FOR

Permite repetir una acción un número determinado de veces. A continuación se dice que repita la sentencia desde el elemento 1 al 10 tomados de

```
INTEGER @counter
 FOR  @counter = 1 TO 10 STEP 2
   DATA() = RESULT LOO KUP() * 0.5
 ENDFOR
```

BREAK

Para el bucle FOR/FOREACH cuando se da cierta condición

```
FOREACH [d/Date]
  IF RESULT LOO KUP ([d /Ac count] = " Tas a") = 0
     BREAK
     ENDIF
  DATA([ d/A ccount] = " Pro por cio n") = RESULT LOO KUP ([d /Ac count] = " Can tid ad") / RESULT LOO KUP (
c count] ="Ta sa")
 ENDFOR
```

Condicionales

IF

Permite ejecutar la sentencia cuando se cumplan ciertas condiciones. Se cierra con ENDIF

```
IF RESULT LOO KUP ([d /Ac count] = " PRI CE") > 0 THEN
 DATA([  d/ E N TI  TY]  ="A s i aP  aci  fi c ", [d/Pro  du c t ]=  "  Pr o d uc  t1")=100
 ENDIF
```

ELSE/ELSEIF

Podemos establecer también qué hacer en el caso en que no se dé la condición declarada en el IF

```
IF [d/ACC OUN T]= " PRI CE" THEN
  ELSEIF RESULT LOO KUP ([d /AC COU NT] ="PR ICE " )>0 THEN
  ELSE
 DATA([ d/E NTI TY] ="As iaP aci fic ", [d/Pro duc t]= " Pro duc t1")=100
 ENDIF
```


Sponsored by ApolloPad.com
Everyone has a novel in them. Finish Yours!
https://apollopad.com