Sum Unique Counts

# Reference documents

<http://vnu10.blogspot.com/2011/01/mdx-grand-total-sub-total.html>

<https://blog.crossjoin.co.uk/2013/05/29/aggregating-the-result-of-an-mdx-calculation-using-scoped-assignments/>

# Step 1

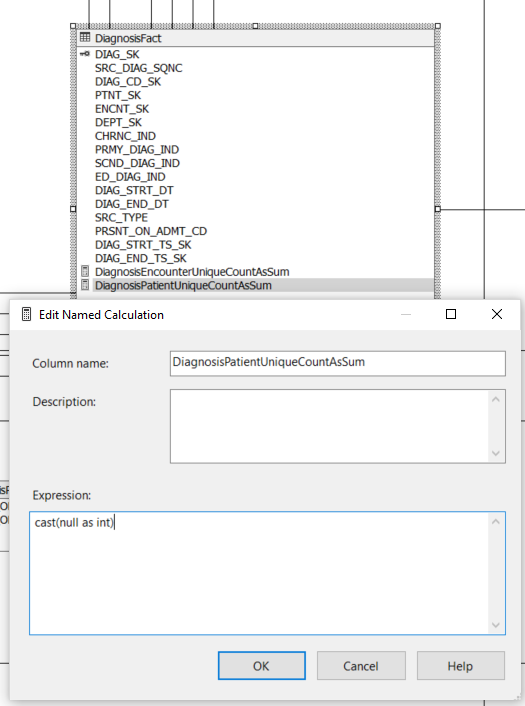
Inside the DSV, in each fact table, create a Named Calculations for each distinct count.

DiagnosisFact has two distinct counts:

* [Diagnosis - Encounter - Unique Count]
* [Diagnosis - Patient - Unique Count]

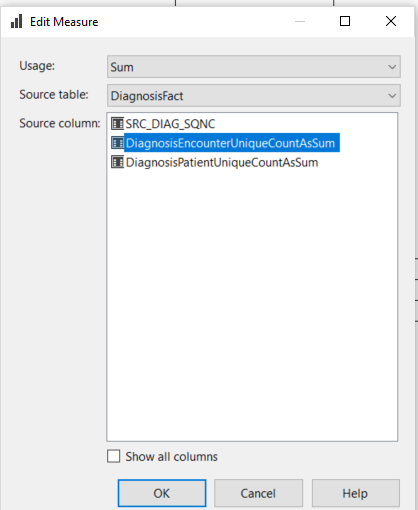
We need to create a Named Calculation for each (please choose your own naming convention):

* DiagnosisEncounterUniqueCountAsSum
* DiagnosisPatientUniqueCountAsSum

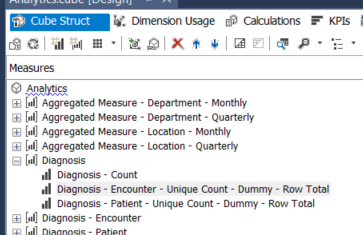


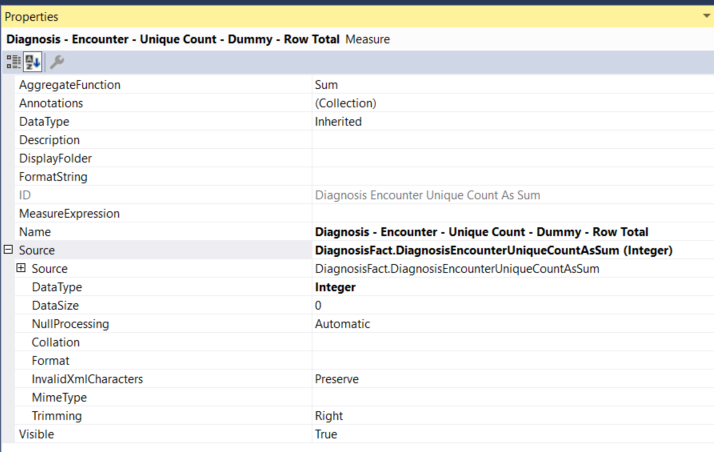
The “Expression” accepts any valid SQL statement. The CAST is used to convert the NULL to the desired data type.

# Step 2



Create a new measure based on the measure in the DSV. Ensure the data type is correct and the measure is **NOT VISIBLE**.





# Step 3

Execute the stored procedure [dbo].[usp\_MeasureGenerator\_UniqueCountRowTotal\_UsingScope] to generate the scope statements.

NOTE: The stored procedure needs a standardized naming convention. Without a standardized naming convention the stored procedure will not be able to determine a relationship between *[Measures].[Diagnosis - Encounter - Unique Count]* and *[Measures].[Diagnosis - Encounter - Unique Count - Dummy – Row Total]*.

# Step 4

Add the new scope to statements to the cube and deploy. Reload the cube the cube metadata.

# Step 5

Test the new measures. The test queries can be generated using the stored procedure [dbo].[ usp\_MeasureGenerator\_TestQueries]. Simply modify the stored to look for the new dummy measures.

SELECT DISTINCT

m.cube\_name

,m.measure\_unique\_name

FROM

dbo.tblCube\_Measures m

where

m.cube\_name = N'analytics'

and (

m.measure\_name like '%dummy%'

)

# Step 6

Execute the stored proc to create the percentage of total [dbo].[usp\_MeasureGenerator\_PercentageOfTotal\_stj].