2020-02-06:

Test:

Currency:

[curr].[currency\_rate] is split to

[curr].[currency\_rate\_external]

[curr].[currency\_rate\_ppl]

insert into [curr].[currency\_rate\_external] (

[currency\_set],[value],[type],[from\_date],[created\_by],[created\_on])

SELECT [currency\_set],[value],[type],[from\_date],[created\_by],[created\_on]

FROM [PricingService].[curr].[currency\_rate] where [type] = 'external'

insert into [curr].[currency\_rate\_ppl] (

[currency\_set],[value],[type],[from\_date],[created\_by],[created\_on])

SELECT [currency\_set],[value],[type],[from\_date],[created\_by],[created\_on]

FROM [PricingService].[curr].[currency\_rate] where [type] = 'ppl'

Review impacted objects:   
SELECT DISTINCT

o.name AS Object\_Name,

o.type\_desc

FROM sys.sql\_modules m

INNER JOIN

sys.objects o

ON m.object\_id = o.object\_id

WHERE m.definition Like '%\[currency\_rate\]%' ESCAPE '\'

Create the tables from existing content

Create a view [curr].[v\_currency\_rate] to replace [curr].[currency\_rate]

CREATE VIEW [curr].[v\_currency\_rate]

AS

select [id],[client],[currency\_set],[value],'ppl'[type],[from\_date],[created\_by],[created\_on] from[curr].[currency\_rate\_ppl]

union all

SELECT [id],null,[currency\_set],[value],'external'[type],[from\_date],[created\_by],[created\_on] FROM [curr].[currency\_rate\_external]

GO

Reassign indexes   
Change in   
C:\inetpub\wwwroot\dba2\app\chart\_js\_srv\_get.aspx.cs

Change in view [curr].[v\_getLatest] Appendix 1.1  
curr.currency\_rate > curr.v\_currency\_rate

Change in API   
C:\inetpub\wwwroot\api\pim\_list\_price.aspx.cs  
(Deploy existing)  
[curr].[currency\_rate] to [curr].[currency\_rate\_ppl]  
Remove: WHERE (curr.currency\_rate\_ppl.type = 'PPL') AND

SQL server jobs:  
job\_once\_a\_week change currency\_rate to currency\_rate\_external,

Also remove [type] and ‘External’

price\_list to \_ppl

Rename old table   
currency\_rate to del\_currency\_rate

Test views, API, sp jobs

DONE!

**Appendix 1.1**

alter VIEW [curr].[v\_getLatest]

AS

WITH TopRows AS (

SELECT curr.currency\_set.from\_curr

,curr.currency\_set.to\_curr

, case curr.v\_currency\_rate.type when 'PPL' then value end as PPL , case curr.v\_currency\_rate.type when 'External' then value end as 'External' , curr.v\_currency\_rate.value

, case curr.v\_currency\_rate.type when 'PPL' then curr.v\_currency\_rate.from\_date end as from\_date

,ROW\_NUMBER() OVER (

PARTITION BY curr.currency\_set.from\_curr, curr.v\_currency\_rate.type,curr.currency\_set.to\_curr

ORDER BY curr.v\_currency\_rate.from\_date DESC

) AS [ROW NUMBER]

FROM curr.currency\_set INNER JOIN curr.v\_currency\_rate ON curr.currency\_set.name = curr.v\_currency\_rate.currency\_set

where type in('PPL','External') and to\_curr <> 'EUR'

)

SELECT from\_curr, to\_curr, sum(PPL) PPL, sum([External])[External],cast(100\*(1-(sum([External])/sum(PPL))) as int) as Diff, max(from\_date )from\_date

FROM TopRows

WHERE TopRows.[ROW NUMBER] = 1

group by from\_curr, to\_curr

GO

**Appendix** 1.2

ALTER PROCEDURE [server].[job\_once\_a\_week]

-- Add the parameters for the stored procedure here

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

SET XACT\_ABORT ON;

insert into [server].[scheduled\_jobs] ([name],[begin\_time]) values ('job\_once\_a\_week',getdate());

DECLARE @identity int;

set @identity = @@IDENTITY;

/\*\*\*\*\*\* code starts here \*\*\*\*\*\*/

/\*\*\*\*\*\* PAT SALES \*\*\*\*\*\*/

update [service\_reference].[sales\_items] set ltm\_qty = null, ltm\_or = null, ltm\_price = null, ltm\_2s\_dev=null;

with cte as (

SELECT service\_reference.sales\_items.item\_number

, SUM(sales.PAT.Qty) AS Qty

, SUM(sales.PAT.[OR]) AS [OR]

, CASE WHEN isnull(SUM(sales.PAT.Qty),0) <> 0 THEN SUM(sales.PAT.[OR])/SUM(sales.PAT.Qty) END AS ltm\_price

, CASE WHEN isnull(SUM(sales.PAT.Qty),0) <> 0 THEN stdev( (sales.PAT.[OR])/(sales.PAT.Qty)) END AS ltm\_2s\_dev

FROM service\_reference.sales\_items INNER JOIN

sales.PAT ON service\_reference.sales\_items.item\_number = sales.PAT.Item\_No

where sales.PAT.[Order\_Date]>cast((year(getdate()) ) \* 10000 + month(getdate()) \* 100 + day(getdate()) as int)-10000

GROUP BY service\_reference.sales\_items.item\_number

)

update [service\_reference].[sales\_items] set

ltm\_qty = cte.Qty, ltm\_or=cte.[OR],ltm\_price = cte.ltm\_price, ltm\_2s\_dev = cte.ltm\_2s\_dev

from cte,[service\_reference].[sales\_items] where cte.item\_number = [service\_reference].[sales\_items].item\_number;

/\*\*\*\*\*\* PAT CURR \*\*\*\*\*\*/

SET XACT\_ABORT ON;

DROP TABLE IF EXISTS #patcurr;

SELECT TOP (100) PERCENT 'EUR-' + Currency\_Code Currency\_Code, AVG(Currency\_Rate\_To\_Euro) AS Avg, MAX(Currency\_Rate\_From\_Date) AS Currency\_Rate\_From\_Date

into #patcurr FROM curr.pat\_dw\_Currency\_Rate

WHERE (Currency\_Rate\_From\_Date > CONVERT(DATETIME, '2008-01-01 00:00:00', 102)) AND (Rate\_Type = N'Variable') AND (Currency\_Code IN

(SELECT to\_curr

FROM curr.currency\_set))

GROUP BY DATEPART(ww, Currency\_Rate\_From\_Date), DATEPART(yyyy, Currency\_Rate\_From\_Date), Currency\_Code

ORDER BY Currency\_Code, DATEPART(yyyy, Currency\_Rate\_From\_Date), DATEPART(ww, Currency\_Rate\_From\_Date);

insert into [PricingService].[curr].[currency\_rate\_external] ([currency\_set],[value],[type],[from\_date]) (

SELECT #patcurr.Currency\_Code, #patcurr.Avg, 'External' , #patcurr.Currency\_Rate\_From\_Date

FROM #patcurr LEFT OUTER JOIN

curr.currency\_rate\_external ON #patcurr.Avg = curr.currency\_rate\_external.value AND #patcurr.Currency\_Rate\_From\_Date = curr.currency\_rate\_external.from\_date

WHERE (curr.currency\_rate\_external.currency\_set IS NULL)

)

/\*\*\*\*\*\* code ends here \*\*\*\*\*\*/

update [server].[scheduled\_jobs] set end\_time = getdate() where id = @identity

END