

Advanced Concepts with Power XL

Agenda

- Import feature
- Custom Layouts
- Form Mode
- Calculation Options with Power XL (SQL vs DAX vs Excel formulas)
- Special Formulas (sparkline, image, barcode)

Import Feature

Common Questions

- Can I import my current Excel file into Power XL?
 - Yes, the import itself will work with any kind of Excel
- Will it import all the sheets?
 - Yes, the import functionality imports all the sheets
- Will it import the formatting and the formulas?
 - Yes, almost all Excel formulas and formatting (e.g.: conditional formatting) are supported
- Will the writeback work?
 - The **writeback will work only on one sheet** (called the data sheet)
 - In order for the import to find the connection between the data model and the Excel's layout, the **following MUST be followed:**
 - Name the **table as "Data"**
 - Use the **exact same column names** as in your model

Import Feature

Configuring import

- **Important**: for the writeback it only matters what you have on the “blue data sheet”
- When creating the Excel for the import, make sure to have a table which is explicitly named as “Data”

The screenshot shows the Microsoft Excel interface with the Table Design ribbon selected. The 'Table Name' field is set to 'Data'. A yellow callout box points to the 'Table Name' field with the text 'Table must be named as "Data"'. The table below has the following data:

	Product	Quantity	Due Date	Design	Build	Ship	Progress	Paint Colo	Order Link	Ship To
1										
2	Personalised Notebook	70	21 Mar. 2022	Delayed	Waiting for Approval	Working on it	500%	#7030A0	AUG0001	235 Bowery, New York, NY 10002
3	Leather Journal	500	19 Apr. 2022	Waiting for Approval	Delayed	Working on it	79%	#fed967	AUG0002	558 Broadway, New York, NY 10013
4	Vintage Necklace	800	16 Oct. 2021	Done	Waiting for Approval	Done	3000%	#70ad47	AUG0003	215 Centre St, New York, NY 10014
5	New item...									

Import Feature

Configuring import

- In order for the writeback to work, the columns of the **Data Table must match the column names** in the data model, otherwise the connection won't be established between the Excel template and the data model

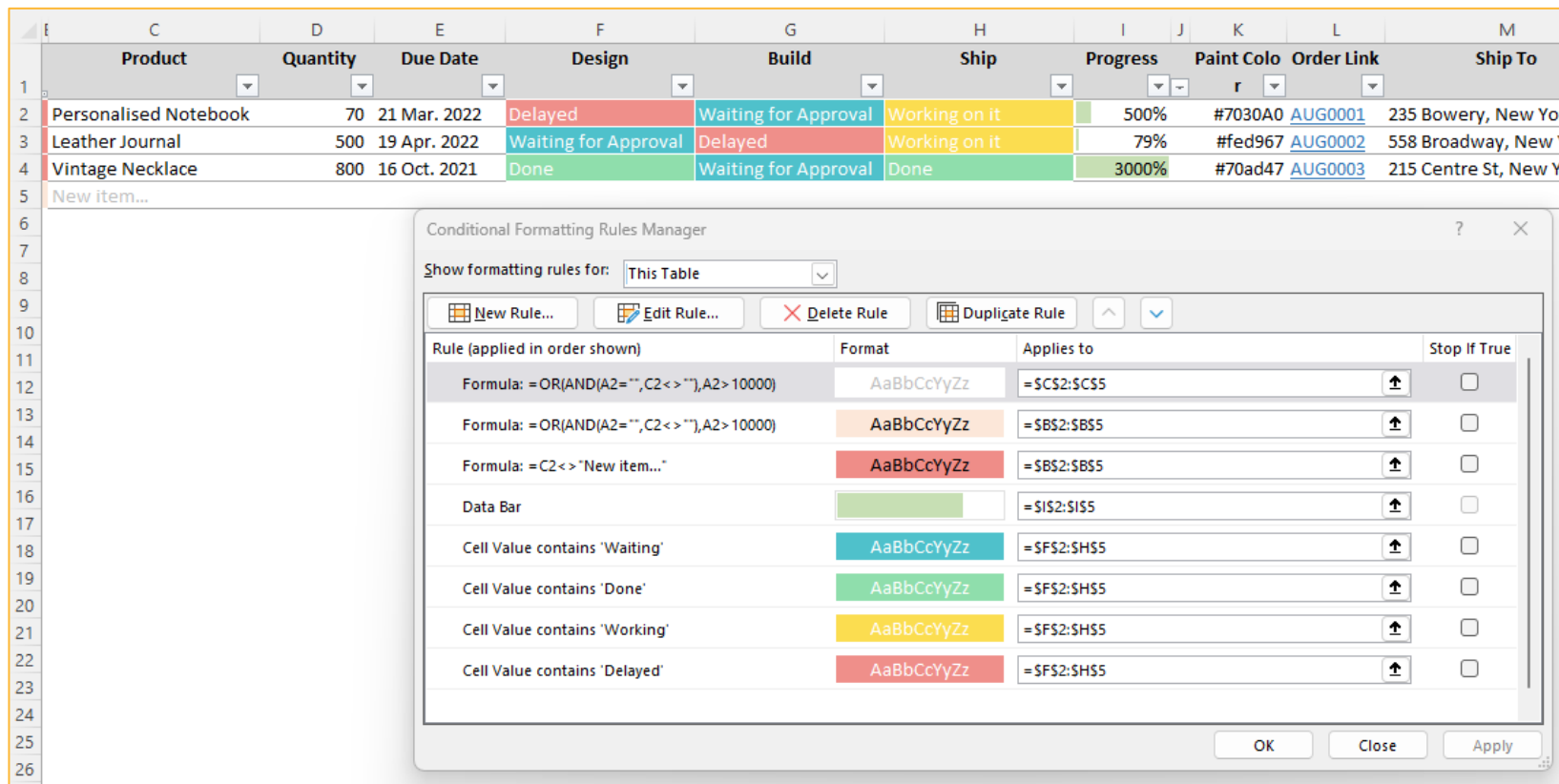
The screenshot illustrates the configuration of an Excel table for data import. The table has columns: Product, Quantity, Due Date, Design, Build, Ship, Progress, Paint Color, Order Link, and Ship To. The 'Table Design' ribbon is active, showing the 'Table Style Options' section. A dropdown menu is open, showing a list of columns from the data model. The columns in the table and the data model are: Product, Quantity, Due Date, Design, Build, Ship, Progress, Paint Color, Order Link, and Ship To. The dropdown menu shows the following columns: ProductOrder, Build, Delivery Date, Design, Due Date, IsDefaultProductOrder, Order Link, OrderSortOrder, Paint Color, Product, ProductOrderId, Progress, Quantity, Ship, and Ship To. Arrows point from the table columns to the corresponding items in the dropdown menu.

Product	Quantity	Due Date	Design	Build	Ship	Progress	Paint Color	Order Link	Ship To
Personalised Notebook	70	21 Mar. 2022	Delayed	Waiting for Approval	Working on it	500%	#7030A0	AUG0001	235 Bowery, Ne
Leather Journal	500	19 Apr. 2022	Waiting for Approval	Delayed	Working on it	79%	#fed967	AUG0002	558 Broadway, N
Vintage Necklace	800	16 Oct. 2021	Done	Waiting for Approval	Done	3000%	#70ad47	AUG0003	215 Centre St, N
New item...									

Import Feature

Configuring import

- Formatting can be applied and will be imported



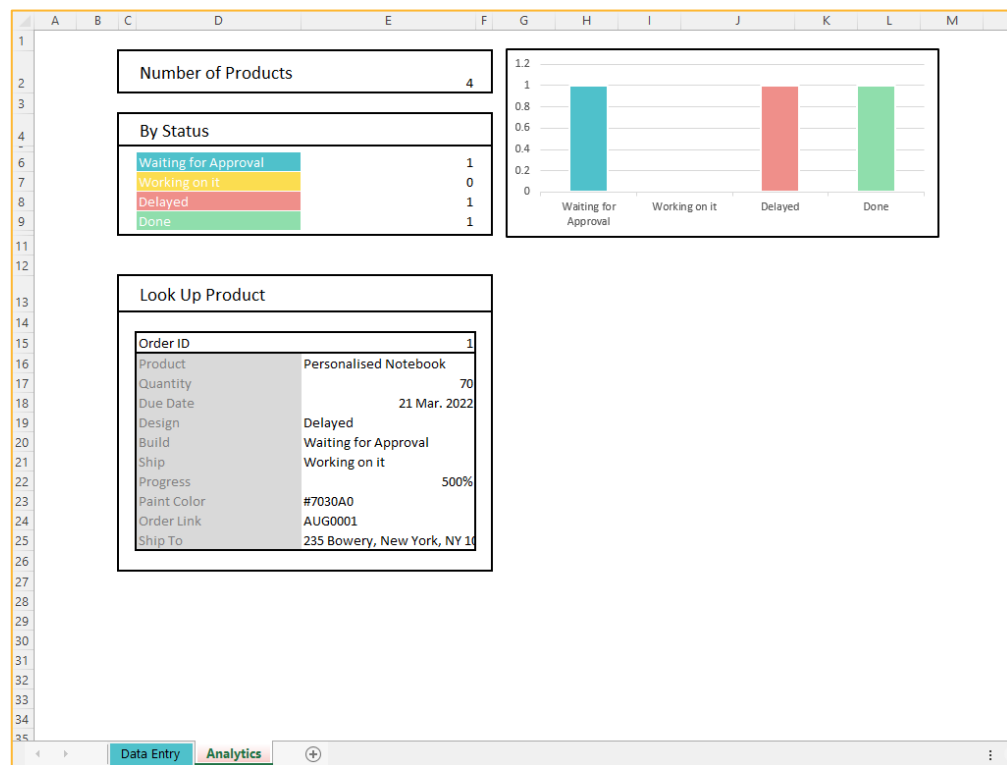
Product	Quantity	Due Date	Design	Build	Ship	Progress	Paint Color	Order Link	Ship To
Personalised Notebook	70	21 Mar. 2022	Delayed	Waiting for Approval	Working on it	500%	#7030A0	AUG0001	235 Bowery, New York
Leather Journal	500	19 Apr. 2022	Waiting for Approval	Delayed	Working on it	79%	#fed967	AUG0002	558 Broadway, New York
Vintage Necklace	800	16 Oct. 2021	Done	Waiting for Approval	Done	3000%	#70ad47	AUG0003	215 Centre St, New York
New item...									

Rule (applied in order shown)	Format	Applies to	Stop If True
Formula: =OR(AND(A2="",C2<>""),A2>10000)	AaBbCcYyZz	=SC\$2:SC\$5	<input type="checkbox"/>
Formula: =OR(AND(A2="",C2<>""),A2>10000)	AaBbCcYyZz	=SB\$2:SB\$5	<input type="checkbox"/>
Formula: =C2<>"New item..."	AaBbCcYyZz	=SB\$2:SB\$5	<input type="checkbox"/>
Data Bar		=S\$2:S\$5	<input type="checkbox"/>
Cell Value contains 'Waiting'	AaBbCcYyZz	=SF\$2:SH\$5	<input type="checkbox"/>
Cell Value contains 'Done'	AaBbCcYyZz	=SF\$2:SH\$5	<input type="checkbox"/>
Cell Value contains 'Working'	AaBbCcYyZz	=SF\$2:SH\$5	<input type="checkbox"/>
Cell Value contains 'Delayed'	AaBbCcYyZz	=SF\$2:SH\$5	<input type="checkbox"/>

Import Feature

Configuring import

- Additional sheets can be added and will be imported as well
- **None of the additional sheets will be writeback capable**, only the Data Table is writeback capable



Import Feature

Configuring import

- Power XL has many non-Excel features, such as lookup from a dimension table (aka Selection column type), special input controls (such as color picker, hierarchy picker) or features related to writeback (such as write security, read only column, key column)
- These settings, which are exclusively available only in Power XL, **can NOT be imported**, these has to be set manually after the import

C Product	Quantity	Due Date	Design	Build
Personalised Notebook	20	18 Mar. 2022	Working on it	Waiting for Approval
Leather Journal	20	19 Apr. 2022	Waiting for Approval	Delayed
Vintage Necklace	400	15 Oct. 2022	Delayed	Waiting for Approval
Scandinavian Print	800	22 Sep. 2021	Working on it	Waiting for Approval
Blue Wall Art	998	31 Aug. 2021	Working on it	
Christmas Tree Kit	70	01 Aug. 2021	Working on it	
Miniature Dollhouse	100	19 May. 2022	Done	
Golden Hawk statue	120	11 Aug. 2021	Working on it	working on it
New item...				

Non-Excel input controls can't be imported

Writeback related settings can't be imported

DESIGN

<input type="checkbox"/> Enable Add / Delete Row Buttons	Column Type	Source Table	<input checked="" type="checkbox"/> Key Column	<input type="checkbox"/> Exclude from save
<input type="checkbox"/> Always Show Insert Row	Text	Display Column	<input type="checkbox"/> Required	
<input type="checkbox"/> Fetch All Data Before Load		Value Column	<input checked="" type="checkbox"/> Read only	<input checked="" type="checkbox"/> Write Security

Column Properties

Import Feature

Configuring import

- During the import the **data from the Data Table is not imported**
- The data is then pulled from the data in the data model, in case Power XL is able to bind the fields (i.e.: the fields were named correctly)

	C	D	E	F	G	H	I	J	K	L	M
	Product	Quantity	Due Date	Design	Build	Ship	Progress	Paint Colo	Order Link	Ship To	
1											
2	Personalised Notebook	70	21 Mar. 2022	Delayed	Waiting for Approval	Working on it	500%	#7030A0	AUG0001	235 Bowery, New York, NY 100	
3	Leather Journal	500	19 Apr. 2022	Waiting for Approval	Delayed	Working on it	79%	#fed967	AUG0002	558 Broadway, New York, NY 1	
4	Vintage Necklace	800	16 Oct. 2021	Done	Waiting for Approval	Done	3000%	#70ad47	AUG0003	215 Centre St, New York, NY 1	
5	New item...										

Data in the import file

	C	D	E	F	G	H	I	J	K	L	M
	Product	Quantity	Due Date	Design	Build	Ship	Progress	Paint Color	Order Link	Ship To	
1											
2	Personalised Notebook	20	3/18/2022	Working on it	Waiting for Approval	Working on it	50%	#fed967	AUG0001	235 Bowery, New York, NY 100	
3	Leather Journal	20	4/19/2022	Waiting for Approval	Delayed	Working on it	79%	#fed967	AUG0002	558 Broadway, New York, NY 1	
4	Vintage Necklace	400	10/15/2022	Done	Waiting for Approval	Done	39%	#70ad47	AUG0003	215 Centre St, New York, NY 1	
5	Scandinavian Print	800	9/22/2021	Done	Waiting for Approval	Delayed	44%	#fed967	AUG0004	175 5th Ave, New York, NY 100	
6	Blue Wall Art	998	8/31/2021	Delayed	Working on it	Waiting for Approval	73%	#2f5697	AUG0005	475 Broadway, New York, NY 1	
7	Christmas Tree Kit	70	8/1/2021	Working on it	Waiting for Approval	Working on it	82%	#c85b11	AUG0006	29 E 4th St, New York, NY 100	
8	Miniature Dollhouse	100	5/19/2022	Done	Delayed	Delayed	55%	#7030A0	AUG0007	252 1st Ave., New York, NY 100	
9	Golden Hawk statue	120	8/11/2021	Working on it	Working on it		22%	#f7ccae			
10	New item...										

Data in Power XL after the import

Custom Layouts

Requirements

- In many cases clients would like to **writeback on a custom layout**
- These custom layouts usually require the following:
 - Some kind of **pivoting** pattern
 - Heavy usage of **conditional formatting, sparklines** or other ways to show certain rules
 - Requirement to **edit the data in bulk**

Custom Layouts

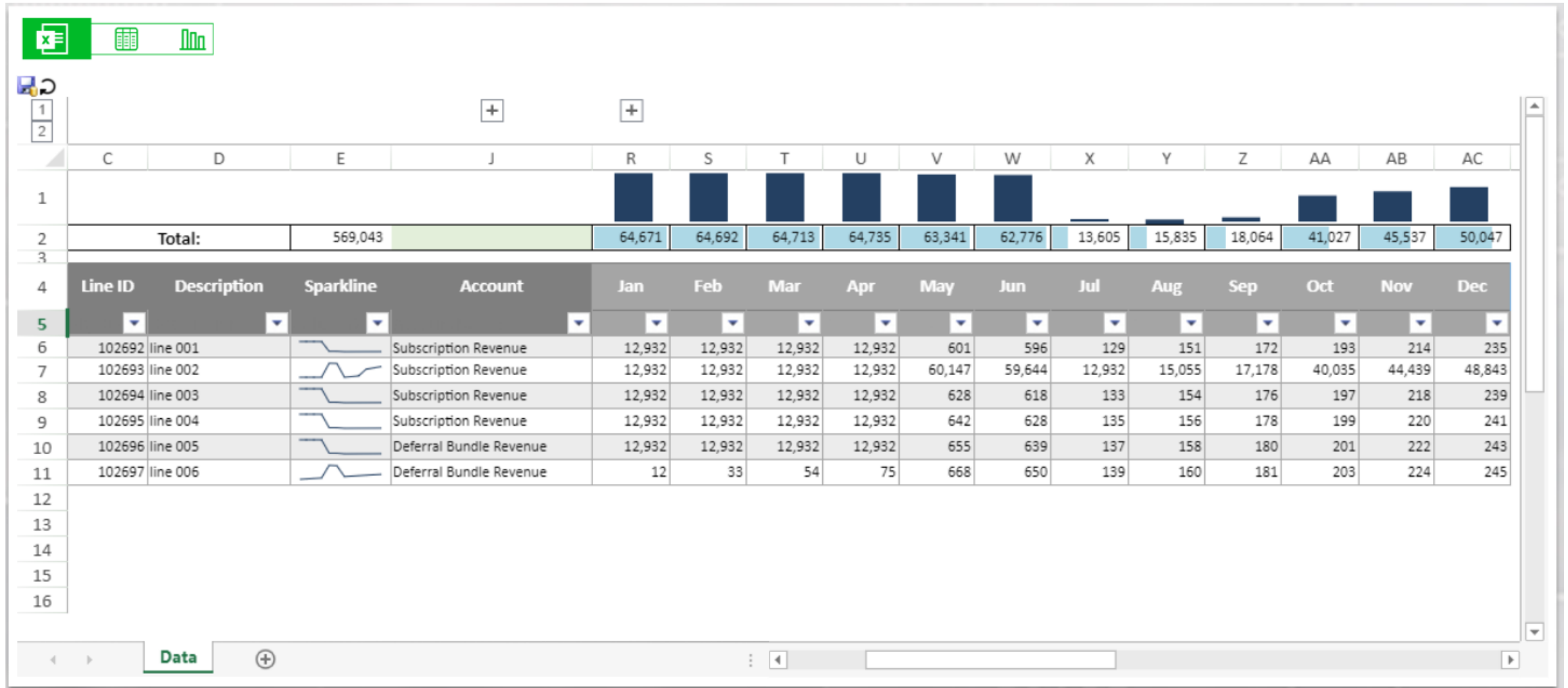
Examples

Edit Projected Quantities 2023-01

1	Project Name	Comment	Planned Quantity	Shipped Quantity	Remaining Quantity	Projected Quantity	Remaining to be Projected	Jan-23	Feb-23	Mar-23	Apr-23	May-23	Jun-23	Jul-23	Aug-23	Sep-23	Oct-23	Nov-23	Dec-23
2																			
3	PROJ00001		75		75	50	25		10	15	5	5	15						
4	PROJ00002		450	202	248	450	(202)	250	200										
5	PROJ00003		100	54	46	46	-	10	30	6									
6	PROJ00004	test	90	20	70	90	(20)	15	15	15	15	5	25						
7	PROJ00005		86		86	86	-		10	20	10	10	10	26					
8	PROJ00006		600		600	550	50		300	250									
9	PROJ00007		33	6	27	28	(1)	4	4	4	4	4	4	4					
10	PROJ00008		75	32	43	60	(17)	15	15	15	15								
11	PROJ00009		50	12	38	35	3	7	7	7	7	7							
12	PROJ00010		75		75	42	33		7	7	7	7	7	7					
13	PROJ00011		300	262	38	100	(62)	100											
14	PROJ00012		56	5	51	56	(5)	5	5	5	8	8	6	5	7	7			
15	PROJ00013		56	8	48	56	(8)	8	8	6	5	7	7	7	4	4			
16	PROJ00014		300	275	25	200	(175)	200											
17	PROJ00015		300	231	69	50	19	50											
18	PROJ00016		38	19	19	10	9	5	5										
19	PROJ00017		150	32	118	150	(32)	30	40	40	40								
20	PROJ00018		50	15	35	-	35												
21	PROJ00019		50	39	11	-	11												
22	PROJ00020		150	158	(8)	-	(8)												
23	PROJ00021		500		500		500												

Custom Layouts

Examples



Custom Layouts

How does it work?

- The pivoting logic happens within a **SQL view**
- The writeback would obviously fail on such a view, but it is **hijacked with an INSTEAD OF trigger**, in order to translate the content of the INSERTED table into insert, update and delete statements
- The view is exposed for writeback in Power XL
- The Power XL grid can be formatted with any kind of formatting logic as needed (only Excel skills are required)

Custom Layouts

Create View

```
9 CREATE VIEW [dbo].[v_BudgetValue] AS
10
11 SELECT
12     a.ConcatViewKey,
13     a.FK_BudgetLineId,
14     a.FK_YearId,
15     b.Jan,
16     b.Feb,
17     b.Mar,
18     b.Apr,
19     b.May,
20     b.Jun,
21     b.Jul,
22     b.Aug,
23     b.Sep,
24     b.Oct,
25     b.Nov,
26     b.Dec
27 FROM
28 (
29     SELECT
30         CONCAT(y.YearId,'|',b1.BudgetLineId) as 'ConcatViewKey',
31         b1.BudgetLineId as 'FK_BudgetLineId',
32         y.YearId as 'FK_YearId'
33     FROM
34         dbo.[BudgetLine] b1
35         ,dbo.[Year] y
36     ) a
37 LEFT JOIN
38 (
39     SELECT
40
41         CONCAT(f.YearId,'|',f.BudgetLineId) as 'ConcatViewKey',
42         f.YearId as 'FK_YearId',
43         f.BudgetLineId as 'FK_BudgetLineId',
44         sum(CASE WHEN m.[Month] = 1 THEN f.[BudgetValue] else NULL END) as 'Jan',
45         sum(CASE WHEN m.[Month] = 2 THEN f.[BudgetValue] else NULL END) as 'Feb',
46         sum(CASE WHEN m.[Month] = 3 THEN f.[BudgetValue] else NULL END) as 'Mar',
47         sum(CASE WHEN m.[Month] = 4 THEN f.[BudgetValue] else NULL END) as 'Apr',
48         sum(CASE WHEN m.[Month] = 5 THEN f.[BudgetValue] else NULL END) as 'May',
49         sum(CASE WHEN m.[Month] = 6 THEN f.[BudgetValue] else NULL END) as 'Jun',
50         sum(CASE WHEN m.[Month] = 7 THEN f.[BudgetValue] else NULL END) as 'Jul',
51         sum(CASE WHEN m.[Month] = 8 THEN f.[BudgetValue] else NULL END) as 'Aug',
52         sum(CASE WHEN m.[Month] = 9 THEN f.[BudgetValue] else NULL END) as 'Sep',
53         sum(CASE WHEN m.[Month] = 10 THEN f.[BudgetValue] else NULL END) as 'Oct',
54         sum(CASE WHEN m.[Month] = 11 THEN f.[BudgetValue] else NULL END) as 'Nov',
55         sum(CASE WHEN m.[Month] = 12 THEN f.[BudgetValue] else NULL END) as 'Dec'
56     FROM
57         [BudgetValue] f
58     LEFT JOIN
```

A view is created, in which the **pivoting logic** (and all other custom logic required for data display) is encompassed

The main aim of this view is to establish the custom layout by pivoting certain elements of your data

Custom Layouts

Instead of trigger

```

9 CREATE TRIGGER
10     [dbo].[trg_InsertBudgetValue]
11 ON
12     [dbo].[v_BudgetValue]
13 INSTEAD OF INSERT, UPDATE
14 AS
15 BEGIN
16
17     PRINT '--Create temp table--'
18     CREATE TABLE #InsertedTempTable
19     (
20
21         [YearId] [int] NOT NULL,
22         [MonthId] [int] NOT NULL,
23         [BudgetLineId] [int] NOT NULL,
24         [BudgetValue] [float] NULL,
25
26     );
27
28     PRINT '--insert into temp table (12 inserts)--'
29     INSERT INTO #InsertedTempTable SELECT FK_YearId, MonthId,FK_BudgetLineId, Jan FROM inserted, [Month] WHERE [Month] = 1;
30     INSERT INTO #InsertedTempTable SELECT FK_YearId, MonthId,FK_BudgetLineId, Feb FROM inserted, [Month] WHERE [Month] = 2;
31     INSERT INTO #InsertedTempTable SELECT FK_YearId, MonthId,FK_BudgetLineId, Mar FROM inserted, [Month] WHERE [Month] = 3;
32     INSERT INTO #InsertedTempTable SELECT FK_YearId, MonthId,FK_BudgetLineId, Apr FROM inserted, [Month] WHERE [Month] = 4;
33     INSERT INTO #InsertedTempTable SELECT FK_YearId, MonthId,FK_BudgetLineId, May FROM inserted, [Month] WHERE [Month] = 5;
34     INSERT INTO #InsertedTempTable SELECT FK_YearId, MonthId,FK_BudgetLineId, Jun FROM inserted, [Month] WHERE [Month] = 6;
35     INSERT INTO #InsertedTempTable SELECT FK_YearId, MonthId,FK_BudgetLineId, Jul FROM inserted, [Month] WHERE [Month] = 7;
36     INSERT INTO #InsertedTempTable SELECT FK_YearId, MonthId,FK_BudgetLineId, Aug FROM inserted, [Month] WHERE [Month] = 8;
37     INSERT INTO #InsertedTempTable SELECT FK_YearId, MonthId,FK_BudgetLineId, Sep FROM inserted, [Month] WHERE [Month] = 9;
38     INSERT INTO #InsertedTempTable SELECT FK_YearId, MonthId,FK_BudgetLineId, Oct FROM inserted, [Month] WHERE [Month] = 10;
39     INSERT INTO #InsertedTempTable SELECT FK_YearId, MonthId,FK_BudgetLineId, Nov FROM inserted, [Month] WHERE [Month] = 11;
40     INSERT INTO #InsertedTempTable SELECT FK_YearId, MonthId,FK_BudgetLineId, Dec FROM inserted, [Month] WHERE [Month] = 12;
41
42
43     PRINT '--update--'
44     UPDATE
45     [BudgetValue]
46     SET
47     [BudgetValue] = t.BudgetValue
48     FROM
49     [BudgetValue] f
50     INNER JOIN
51     #InsertedTempTable t
52     ON
53     f.[BudgetLineId] = t.[BudgetLineId]
54     AND f.YearId = t.YearId
55     AND f.MonthId = t.MonthId;

```

An **instead of trigger** is created, which hijacks the writeback and translates it to insert/update/delete

The values in the INSERTED virtual table are parsed in this instead of trigger and translated into insert/update/delete statements

Certainly, this view on it's own will never be writeable, hence it is a requirement to implement the instead of procedure on it, which will then perform the insert/update/delete into the underlying table or tables

Custom Layouts

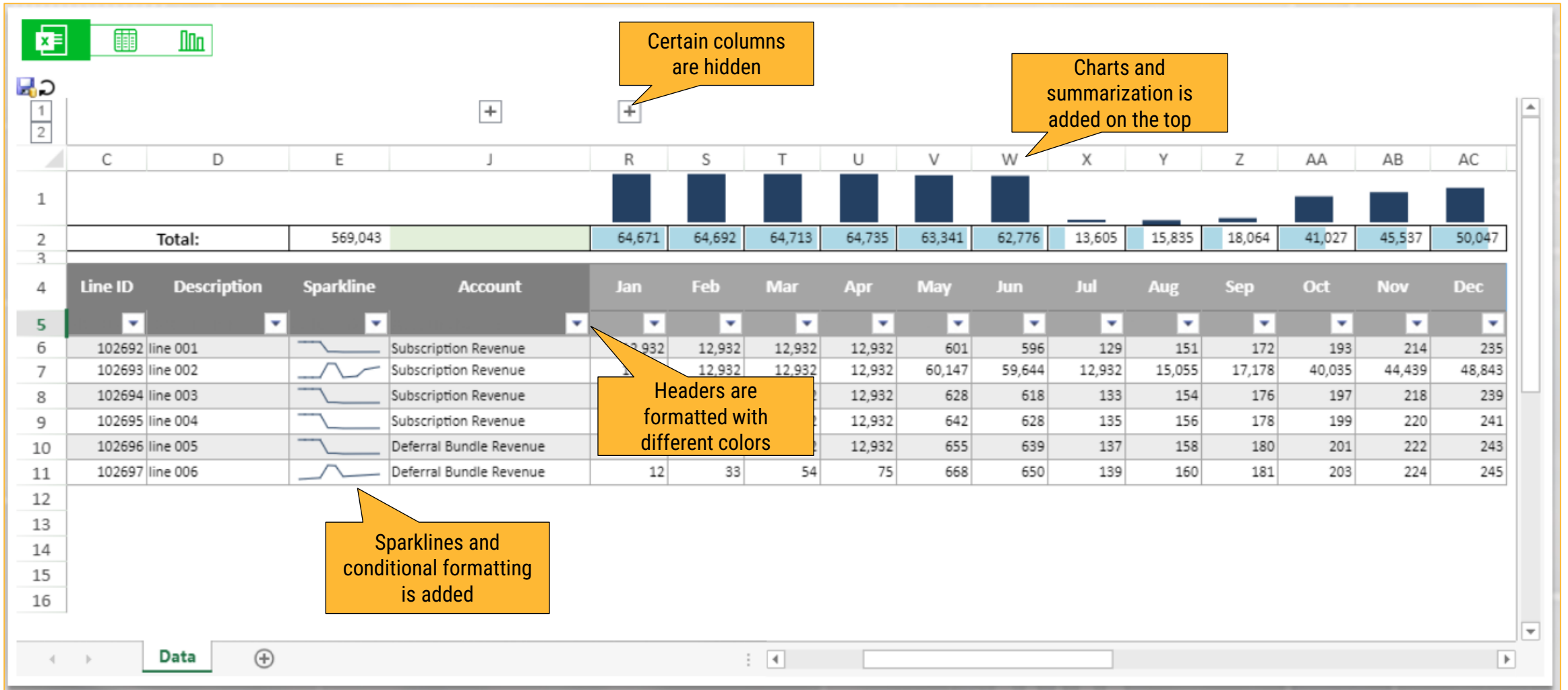
Expose View in Power XL

The view is then exposed in Power XL, so that it can be used for writeback

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	ConcatViewKey	FK_YearId	FK_BudgetLineId	Description	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
2	2 102692	2.00	102,692.00	line 001	12,932	12,932	12,932	12,932	601	596	129	151	172	193	214	235
3	2 102693	2.00	102,693.00	line 002	12,932	12,932	12,932	12,932	60,147	59,644	12,932	15,055	17,178	40,035	44,439	48,843
4	2 102694	2.00	102,694.00	line 003	12,932	12,932	12,932	12,932	628	618	133	154	176	197	218	239
5	2 102695	2.00	102,695.00	line 004	12,932	12,932	12,932	12,932	642	628	135	156	178	199	220	241
6	2 102696	2.00	102,696.00	line 005	12,932	12,932	12,932	12,932	655	639	137	158	180	201	222	243
7	2 102697	2.00	102,697.00	line 006	12	33	54	75	668	650	139	160	181	203	224	245
8																
9																
10																
11																
12																
13																
14																
15																
16																

Custom Layouts

Format Power XL Table



Custom Layouts

Conclusions

- **Pros:**
 - Enable a very **flexible data entry** approach with relatively low effort of coding
 - Can be implemented based on **sample SQL code**
 - Extremely **fast** (way faster than DEMx, because this is not a tuple based operation)
 - Customizable and can leverage many **Excel features**
- **Cons:**
 - Not out of the box, needs some **SQL skillset**
 - Still not ideal for thousands of records to be copy and pasted at the same time (ideal for a few hundreds of rows)
 - Doesn't enable writeback on totals or subtotals

Form Mode

Requirements

- In many cases clients would like to establish a **form like look and feel**
- This enables users to have a **visually appealing layout**, instead of having the rows one by one below each other with VTE

Form Mode

Power XL Form vs VTE Form

TBO Threshold Parameters
94%

General

Segment: Grape

DW User Auth GPID: DUA07787

DW Last Update Time: #####

Min. Sales Exec. Threshold (National): 0

Dimensional Value: 4

Innovation

Innovation Type Refresh: 6

Innovation Type Reframe:

Sales

Sales Green: 47%

Sales Yellow:

Save Changes Discard Changes Insert Delete

Segment

Dimensional Value

Sales Green Percentage

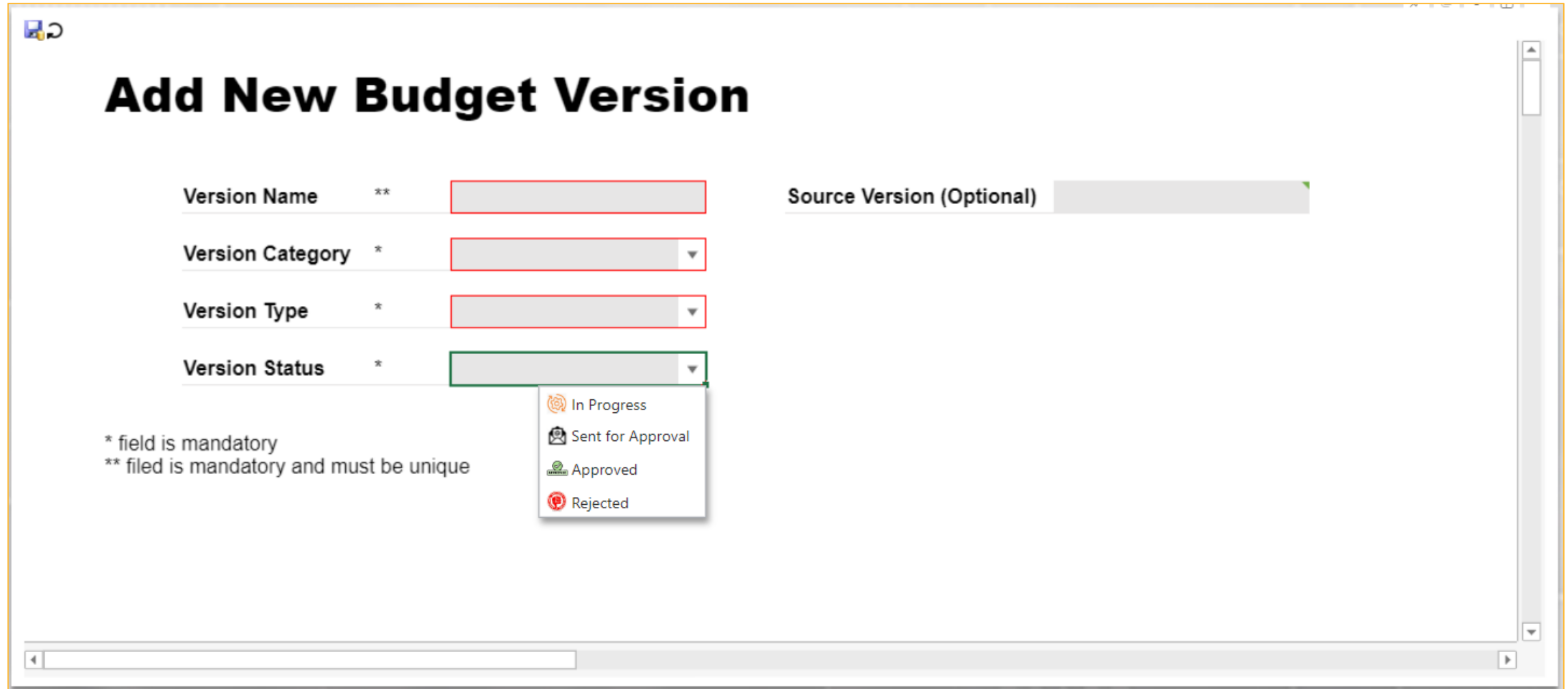
Sales Yellow Percentage

Sales Red Percentage

Margin Green Percentage

Form Mode

Examples



Add New Budget Version

Version Name **

Version Category *

Version Type *

Version Status *

Source Version (Optional)

* field is mandatory
** field is mandatory and must be unique

- In Progress
- Sent for Approval
- Approved
- Rejected

Form Mode

Examples

TBO Threshold Parameters
94%

General

Segment	Grape
DW User Auth GPID	DUA07787
DW Last Update Time	#####
Min. Sales Exec. Threshold (National)	0
Dimensional Value	4

Innovation

Innovation Type Refresh	6	Innovation Type Reframe	9	Innov. Type Breakthr.	18
-------------------------	---	-------------------------	---	-----------------------	----

Sales

Sales Green	47 %	Sales Yellow	75 %	Sales Red	96 %
-------------	------	--------------	------	-----------	------

Form Mode

How does it work?

- Need to create an **extra form sheet**, on which the fields will be aligned in the custom form layout
- Need to reference the **values from the data sheet to the form sheet**, in order to get the initial values of the form
- Then need to reference the **values back from the form sheet to the data sheet** (remember, it only matters what is on the data sheet)

Form Mode

Create form Sheet

The screenshot shows a form titled "TBO Threshold Parameters" with a 94% progress bar. The form is organized into three sections: "General", "Innovation", and "Sales".

- General Section:** Includes fields for "Segment" (Grape), "DW User Auth GPID" (DUA07787), "DW Last Update Time" (#####), "Min. Sales Exec. Threshold (National)" (0), and "Dimensional Value" (4).
- Innovation Section:** Includes fields for "Innovation Type Refresh" (6), "Innovation Type Reframe" (9), and "Innov. Type Breakthr." (18).
- Sales Section:** Includes a "Sales Green" field (47%) with a dropdown menu, and "Sales Yellow" (75%) and "Sales Red" (96%) fields.

Callouts point to various features: "Add a title" points to the form title; "Add a progress bar (with a sparkline formula)" points to the 94% bar; "Add form sections" points to the section headers; "Add icons" points to the list icon; "Align the fields in a meaningful layout" points to the grid layout; "Add form controls" points to a dropdown menu; and "Add conditional formatting" points to the colored progress bars.

Form Mode

Limitations

- Form is **not reset** after save or with slicer change
 - Need to have a page navigation or a bookmark hide/unhide event, in order to have the form refreshed (or click the refresh button)
- **Dynamic dropdowns** are possible, but only with work arounds (DAX and formula tricks, see Power XL Table - Best Practice Documentation on the KB site)
 - Simple hardcoded dropdowns are easy
- **New form is tricky**, in most of the cases requires an instead of trigger in SQL on a technical item (see Power XL Table - Best Practice Documentation on the KB site)

Calculations with Power XL

Overview

- When working with Power ON and Power XL, there are the following ways to let something calculated
 - SQL triggers or stored procedures
 - Power Query (M Code)
 - DAX calculations
 - Excel formulas
- All of these can be used for different purposes

Calculations with Power XL

SQL Triggers and Stored Procedures

- The MS SQL language is very powerful to **run calculations on your data tables**
 - Stored procedures are like functions (can have inputs and outputs) and perform a certain set of actions
 - A trigger is always related to a table and “triggered” on an event (insert, update, delete). Once triggered, it performs a set of operations, invokes stored procedures or lets the operation fail on purpose (if business rules are violated)
- When to use it?
 - When you need to **physically store** the outputs of the calculations (e.g.: create a new version as a copy of a previous version)
 - When you need to **validate business rules**
 - When the calculation is that **complex** or that resource intense, that it is impossible to encompass with DAX (e.g.: recursive or iterative calculations, complex FX based calculations)

Calculations with Power XL

Power Query (M Language)

- Power Query is a data prep language and ideal for typical **ETL** related operations
 - If extensive transformations are used, then writeback won't be supported and the refresh performance will be depreciated
 - Hence use Power Query consciously
- When to use it?
 - If the table is read only and doesn't play a role in writeback
 - When the transformation is complex and hard to do with a SQL view (e.g.: unpivoting)
 - When it is just quicker to do the transformation in Power Query (no code approach)

Calculations with Power XL

DAX

- DAX stands for **Data Analysis Expressions**
 - It is very powerful for creating complex analytical calculations, which can be sliced and diced and reused in different contexts
 - It can encompass complex business logic within a few lines of code and leverage insights
- When to use it?
 - When you need to encompass business logic for **analytics and data visualizations**
 - Data analysis with complex calculations (e.g.: weighted avg, st dev, ratios, distribution)
 - When the results can be interpreted in different contexts (e.g.: YTD calculations, month on month comparisons, allocation, dynamically display in multiple currencies)

Calculations with Power XL

Excel formulas

- Power XL leverages **Excel formulas**
 - It can use almost all Excel formulas
 - However these formulas can **not be delegated** to the back end
 - The visual first needs to pull all the data to the browser of the user and then the formula runs in the user's browser, which can have negative performance implications (especially on bigger data)
- When to use it?
 - When the calculation is easier to be managed with Excel formulas, than with DAX
 - When it is a simple row based operation (e.g.: Quantity * Price = Sales) and you don't want to handle that with DAX or SQL triggers
 - When you wish to show a total, subtotal or a simple chart above the data table, even before the save is committed
 - When the data volume is not huge (won't work quickly with running Excel formulas on top if multiple thousands of records)

Calculations with Power XL

Conclusions

	SQL Stored Proc / Trigger	Power Query (M)	DAX	Excel
When to use?	Store Data Physically	Transforming data, ETL (not for WB tables)	Complex analytical calculations	Simple aggregations on the front end, showing totals before save
Example	Create new version	Unpivoting	YTD, YoY%, St Dev, Weighted Avg	Total and Subtotal on top of the data table
Advantage	Very quick and runs on the server side	Easy to use (no coding needed)	Very powerful for analytics, can be used in multiple context	Very easy to use
Disadvantage	Requires enhanced SQL scripting skills	Can be slow, doesn't support writeback	Hard to master <i>"DAX is simple, but it is not easy"</i>	Run on the front end, can be slow on higher data volume

Special Power XL Formulas

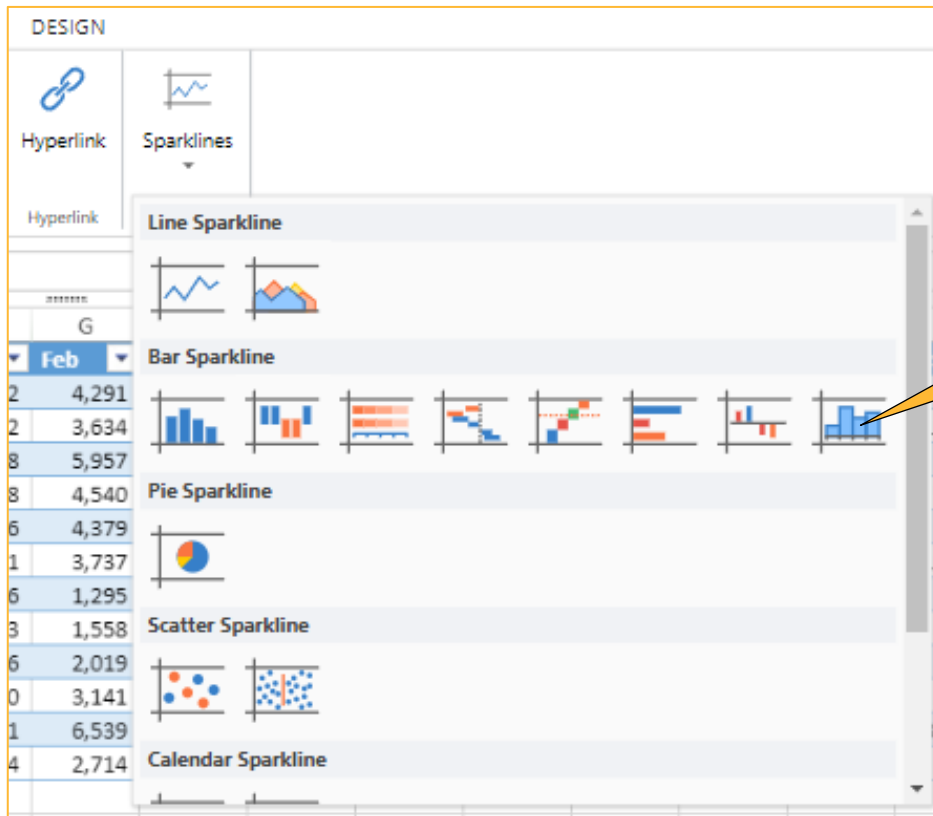
Overview

- In the section we will cover the special Power XL formulas, which are not available as regular Excel formulas, such as:
 - **Sparklines**
 - Sparklines can bring the Power XL reports to the next level
 - Sparklines are basically charts within a cell
 - They can be easily added and will behave as regular formulas
 - **Images**
 - Using the IMAGE() formula, users can resolve an image URL and display that instantaneously within a cell
 - **Barcodes / QR Codes**
 - If you have a URL, you can encode that into a QR code with Power XL
 - Alternatively you can display barcode information in barcode format and read that with a barcode reader

Special Power XL Formulas

Sparklines

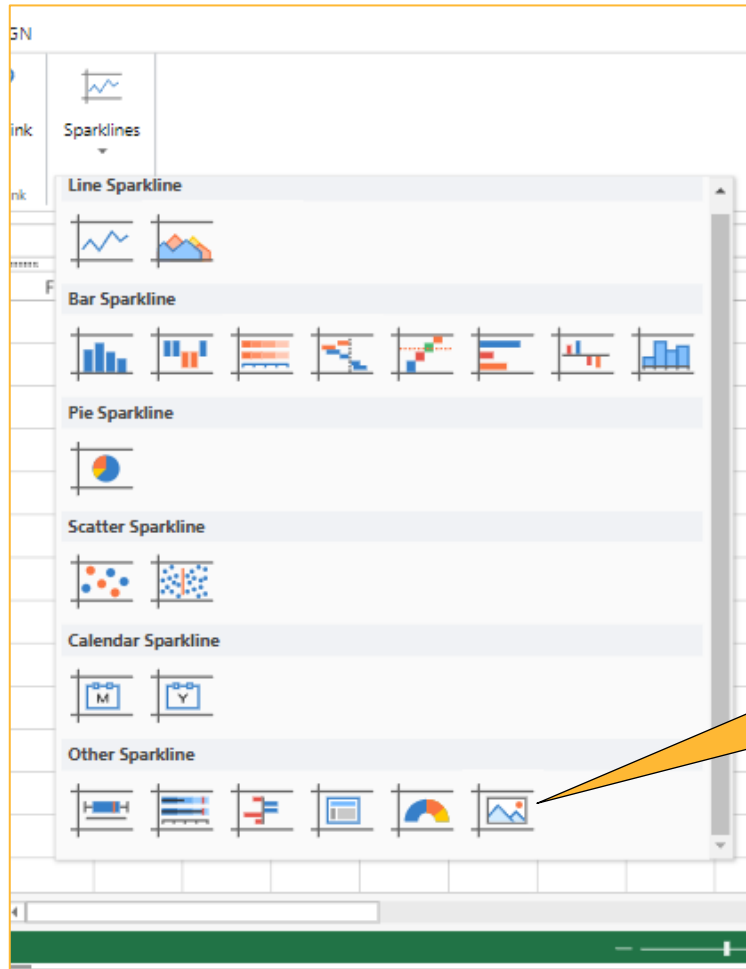
- =LINESPARKLINE(E2:P2,1)
- =COLUMNSPARKLINE(F2:Q2,1)



Browse from different kind of sparklines

Special Power XL Formulas

Images

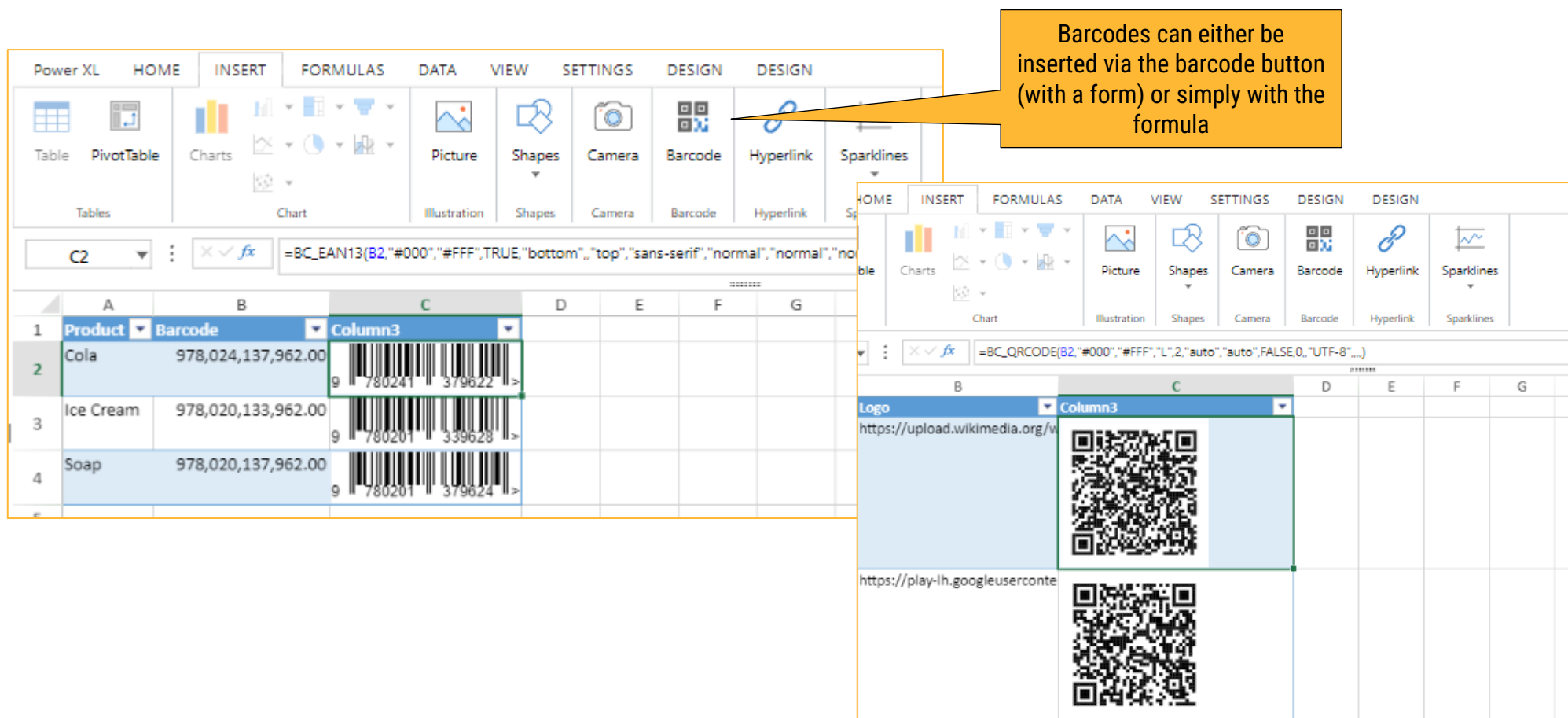


Add image formula and supply it with a URL which points to an image on the public domain




Special Power XL Formulas


QR Codes / Barcodes

- =BC_EAN13(B2,"#000","#FFF",TRUE,"bottom","top","sans-serif","normal","normal","none","center",12,....)
- =BC_QRCODE(B2,"#000","#FFF","L",2,"auto","auto",FALSE,0,"UTF-8",....)



Barcodes can either be inserted via the barcode button (with a form) or simply with the formula

Product	Barcode	Column3
Cola	978,024,137,962.00	
Ice Cream	978,020,133,962.00	
Soap	978,020,137,962.00	

Logo	Column3
https://upload.wikimedia.org/v	
https://play-lh.googleusercontent	