

Anaplan

モデル設計の考え方

August 2022

免責事項

本資料は当社の一般的な製品の内容を概説することを目的としています。

参考情報であり、掲載されている情報は予告なしに変更されることがあります。

情報提供を目的としており、いかなる契約にも組み込む事はできません。

資料、コード、機能を提供することを約束するものではなく、意思決定を下す際に依存するものではありません。

Anaplan社は本資料の内容についていかなる保証も致しません。

また、本資料の内容に関するいかなる損害についても責任を負いかねます。

Anaplanの製品について説明されている機能または機能の開発、リリース、およびそのタイミングはAnaplan社独自の裁量に委ねられます。

本資料とAnaplan社の正式発表に内容が異なる場合は正式発表の内容が優先されます。

基本となる設計思想 P.L.A.N.S

Anaplanは、パッケージシステムではなく「プラットフォーム」ソリューションであるため、機能要件を満たすための構築方法は何通りも存在します。ただし、以下の大原則を念頭において設計・構築を進めることが、パフォーマンス、拡張性、メンテナンス性、といった非機能要件をきちんと満たす優れたソリューションとなるかどうかを左右します。

P.L.A.N.S - 最適なモデル設計

P

Performance

HyperBlockの特徴を活かす構造/数式

L

Logical

リスト・モジュール・ラインアイテムの最適な構成 (DISCO)

A

Auditable

可読性・メンテナンス性を考慮した数式

N

Necessary

不必要な演算や次元を保持しない。計算は1回のみ、それを複数箇所が参照する

S

Sustainable

運用や将来的な拡張性を常に意識する

Anaplanの設計原則 D.I.S.C.O

D.I.S.C.O. – P.L.A.N.Sを実現するための最適なモジュール配置

D

Data

- 外部データ(実績など)の取り込み用モジュール
- Data Hub

I

Inputs

- ユーザーの入力用モジュール
- Pageに配置される

S

System

- 時間管理、画面表示制御用フィルタ、DCAドライバなど制御系モジュール
- マスタマッピング

C

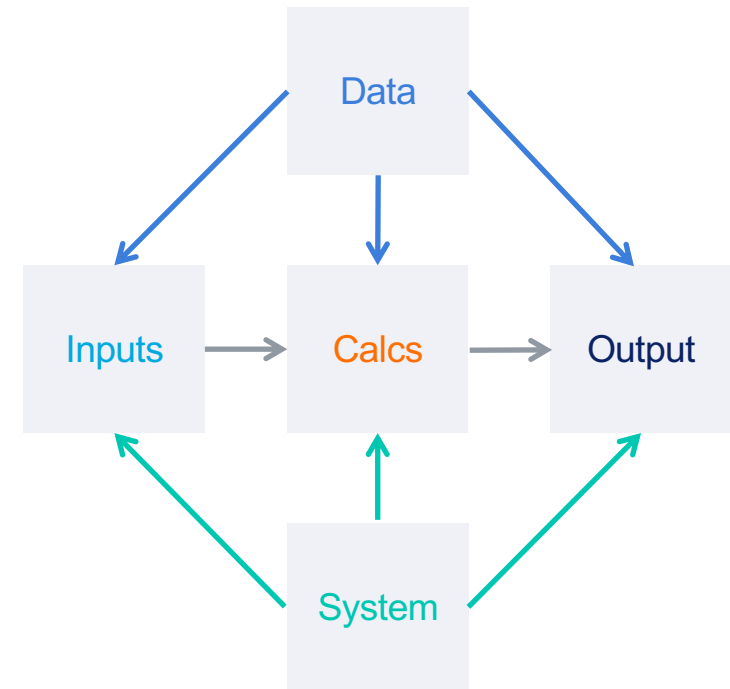
Calcs

- InputやDataの値を用いてロジックを組み込む計算用モジュール

O

Outputs

- InputやCalculation結果を表示する参照モジュール
- Pageに配置される



DataとSystemモジュールは、Input, Calculation, Outputモジュールに参照される

D.I.S.C.Oを守らないことによるデメリット

1つのモジュールになんでもかんでも詰め込んでしまうと、以下のようなデメリットが発生してしまいます。

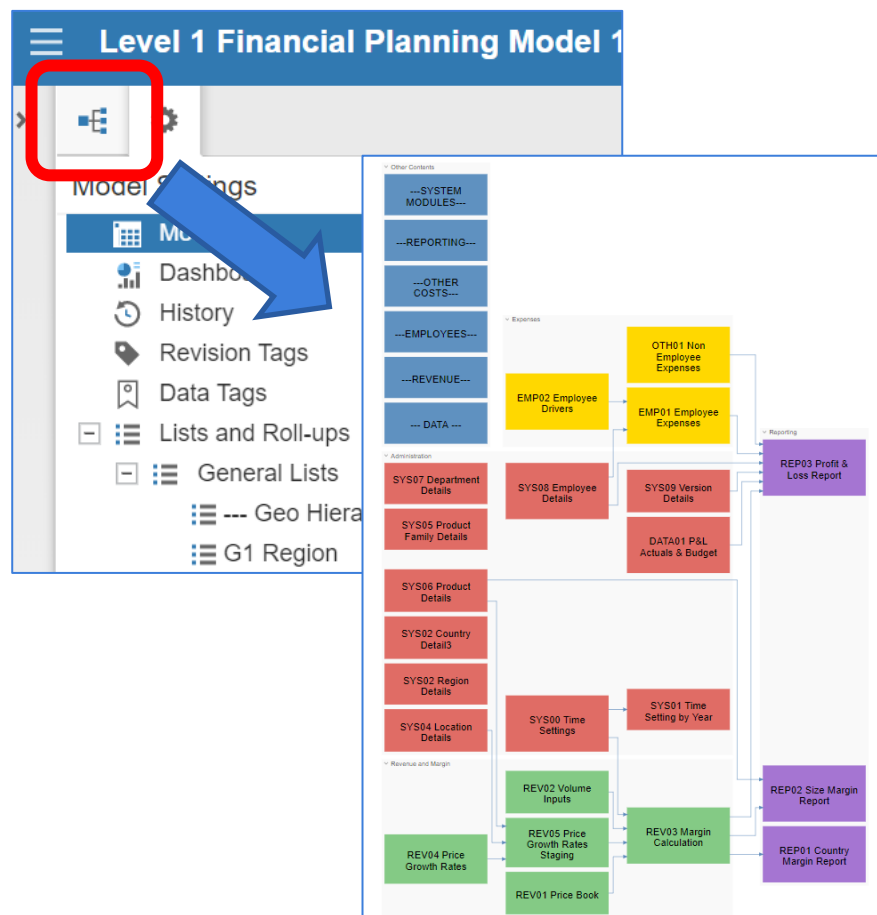
- 同じ式を至る複数回書いてしまうということが発生し、冗長したモデルになる
- 不必要な容量消費が発生する
- どのモジュールになにが入っていたのか分からなくなり、生産性が下がる。
- 誤ってエンドユーザーに見えてはいけない計算結果が見えてしまう。
- モジュールをPageに展開した後に、ユーザーに見せなくていい項目を毎回Hide、Show。
- Model Map上でデータフローが一方向にならない。
- 不要になったLine Itemがどれだかわからなくなる。

D.I.S.C.Oを守ることにによるメリット

モジュールをD.I.S.C.Oに従ってわけていくと以下のようなメリットがあります。

- 同じ式を複数回書いてしまうということが発生しなくなる
- 必要な分のみの容量消費となる
- 変更の影響範囲がわかりやすい
- 機能追加がしやすい
- レビューしやすい
- Page上でのHide、Showなどをしなくて良くなる
- アクセス権の設定が楽になる
- 容量削減時の分析がしやすい
- 不要なLine Itemが一目瞭然なので消せる（複数人開発だと特に効果発揮）

モデルマップ



- モデル内のモジュールとデータの流れが確認出来ます。
- モデルマップ内の色分けはFunctional Areaの設定に依存します。(色は指定不可)
- モデルマップへのアクセス権はモデルビルダーの権限を持ったユーザーのみです。
- モジュールの設定を変更した場合、自動的にモデルマップも更新されます。

→ 開発中はモデルマップを時々確認し、データの流れが左から右に一方向に流れるキレイなデータフローになっているか確認しましょう。

DISCOの推奨命名規則

モジュールの名称にDISCOを含めることで、そのモジュールの用途を名称から理解できるようにする

Anaplan社命名規約書より抜粋

[DISCOの頭文字]_[任意名称]	
[DISCOの頭文字]	D.I.S.C.Oのいずれか1桁 *1桁ではなくIN_、OUT_などでも問題はないが、最初の1桁で識別可能なため本書では1桁とする *1つのモジュール内でIn、Calc、Outが混在する例については次ページで解説する
[任意名称]	そのモジュールの用途が分かる簡潔な任意名称
例	D_Sales Volume Actual I_Sales Volume Forecast C_SalesVolume Actual Allocation O_Sales Variance Analysis
解説	DISCOの頭文字を名称に含めることで可読性が大きくあがる。モジュール名称の末尾に入れると、モジュール一覧画面で見切れる可能性があるため、プリフィックスとして定義することを推奨する。

UI要件によって、In、Calc、Outが混在するケースは必ず発生する。命名規則としては、Input > Output > Calculationの順で優先し、Page上に展開されているモジュールは必ずInもしくはOutとし、展開されていないモジュールのみがCalcとしてモデル内に存在するようにする。

Data	Input	System	Calc	Output	Prefix	Remarks
●					D_	
	●				I_	
		●			S_	
			●		C_	
				●	O_	
	●		●	●	I_	In, Calc, Out混在の場合はI
	●		●		I_	In, Calc混在の場合はI
	●			●	I_	In, Out混在の場合はI
			●	●	O_	Calc, Out混在の場合はO

一方で、中身と見合わない名称をつけてしまうと(Input項目があるのにC_xxxなど)、余計な混乱を生むため、それぞれの位置づけをきちんと理解しそのモジュールがどういった用途かを考えてネーミングすること。

DISCOのどれに該当するか？ 基本の考え方

AppsのPageに展開するモジュール

- 見込値や補正值などのエンドユーザーの入力用モジュールは、Input
- 計算結果やP/Lレポートの表示は、Output
- ユーザーフィルターや管理者のパラメータ値（Current Period等）は、System
- 取り込んだ外部データ（実績データやマスタ情報）は、Data

AppsのPageに展開しないモジュール

- InputからOutputの間の計算を行うものは、Calculation
- 取り込んだ外部データ（実績データやマスタ情報）は、Data
- FilterやDCAの条件になるもの、時間や為替や配賦ドライバの設定は、System

DISCOのどれに該当するか？ 基本の考え方

すべてのケースに当てはまるわけではないが、以下が基本的な位置づけとなる

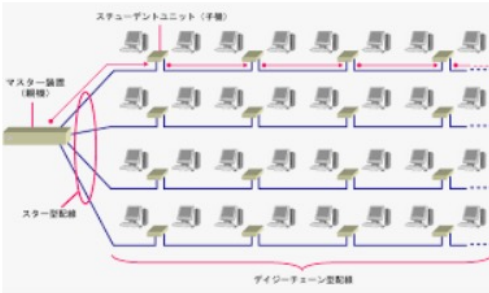
	Data	Input	System	Calculation	Output
User Interaction	Low/Medium	High	Low	None	High
User Type	管理者 エンドユーザー (View only)	エンドユーザー	管理者 エンドユーザー (頻度低)	管理者	エンドユーザー
Summary	Off	On	Off	Off	On
Data flows in	外部システムデータ Data Hub	マニュアル入力 Data Module System Module	マニュアル入力 Data Module	Data Module Input Module System Module	Data Module Input Module System Module Calculation Module
Data flows out	Input Module Calculation Module Output Module System Module	Calculation Module Output Module	Data Module Input Module Calculation Module Output Module	Output Module	Export

<https://community.anaplan.com/t5/Best-Practices/Best-practices-for-module-design/ta-p/35993>

Don't Daisy Chain - デイジーチェーンは止めましょう

Google

デイジーチェーン



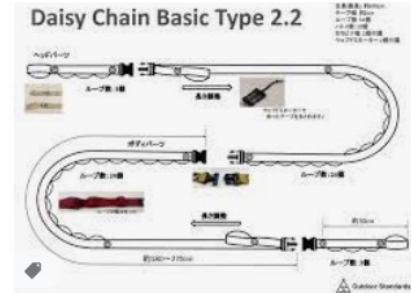
デイジーチェーン方式とスター型混合配線の特長と...
landcomp.co.jp



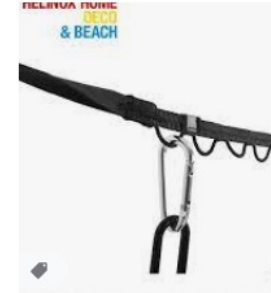
デイジーチェーン | C & C. P. H. E Q...
pitagorabase.com · 在庫あり



アスクル] トーウ (TOWA) TOWA...
askul.co.jp · 在庫あり



デイジーチェーン Basic Type 2.2 <OlivexOl...
outdoorstds.base.shop



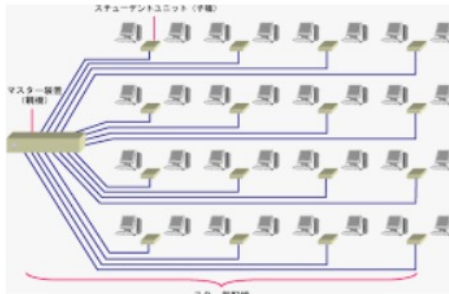
ヘリノックス デイジーチェ...
aandfstore.com



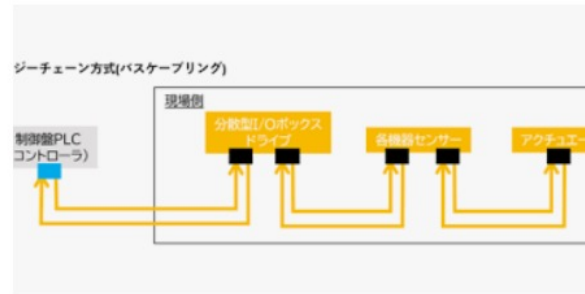
アスクル] トーウ (TOWA) TOWA ...
askul.co.jp · 在庫あり



アルティメイトデイジーチェーン(...
lostarrow.co.jp



デイジーチェーン方式とスター型混合配線の特...
landcomp.co.jp



デイジーチェーン方式でのイーサネット通信
lappjapan.lappgroup.com



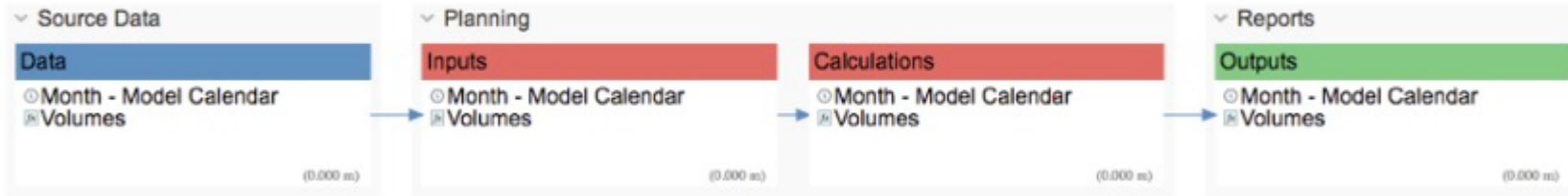
デイジーチェーン掛...
ameblo.jp



Don't Daisy Chain - デイジーチェーンは止めましょう

Daisy Chainの状態

DataモジュールのVolumeをInputモジュールが参照し、それをCalcが参照し、それをOutputが参照している状態

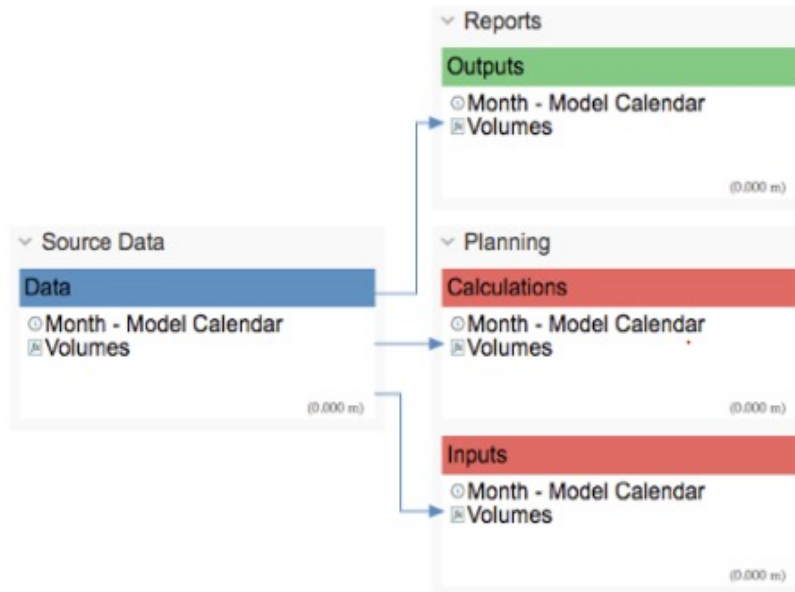


計算エンジンに不可がかり、パフォーマンスに悪影響

※もちろん、Dataに対して何かしらの計算を施した結果をOutputに表示する、ということであればこのデータフローになって全く問題ないです。ここでは全く同じ値を数珠つなぎに参照してしまう状態のことを問題としています。

Daisy Chainではない最適な状態

DataモジュールのVolumeをInput、Calc、Outputがそれぞれが参照している状態



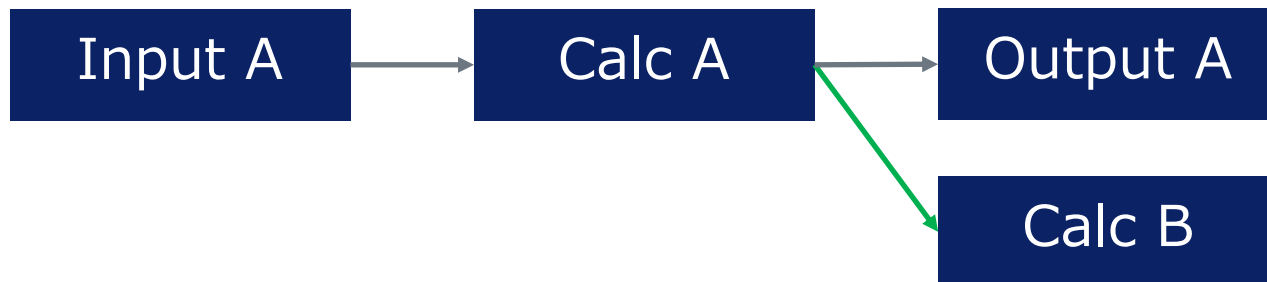
複数箇所が一箇所を参照し、最適な状態

Don't Daisy Chain - デイジーチェーンは止めましょう

Daisy Chainは、パフォーマンスのみならずさらに拡張性を低下させます。

例えば以下の例だと、機能Aの計算結果を機能Bの計算で参照したい場合、Calc BがOutput Aを参照するよりも、Calc BはCalc Aを参照する形にしてあげるほうが、影響範囲調査や仕様変更が容易になります。

例えば、後から「数値の表示は単位を切り替える機能をつけてほしい。」と言われた場合、OutputがCalcから参照されていると機能追加が難しくなります。



System Module

ほとんどのモデル構築において作成することになる代表的なシステムモジュールを紹介します。

Time Filter

Time FilterはAnaplan構築においてまず最初に習得すべきテクニックの一つです。1モデル内に必ず一つTime Filterモジュールを作成するものと理解してください。InputやOutputモジュールのフィルター条件となります。

Systemモジュールとしてモデル内で一元管理し、様々なInput/Outputモジュールが使用します。Input/Outputモジュールそれぞれでフィルター条件を作ることはやめましょう。

OK Systemモジュールで一元管理

SYS00 Time Settings

	Jan 20	Feb 20	Mar 20	Apr 20	May 20	Jun 20	Jul 20	Aug 20
First day of Month	2020/01/01	2020/02/01	2020/03/01	2020/04/01	2020/05/01	2020/06/01	2020/07/01	2020/08/01
Last day of Month	2020/01/31	2020/02/29	2020/03/31	2020/04/30	2020/05/31	2020/06/30	2020/07/31	2020/08/31
Current Period?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Current Period Start	2020/10/01	2020/10/01	2020/10/01	2020/10/01	2020/10/01	2020/10/01	2020/10/01	2020/10/01
Actual Period?	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Forecast Period?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Forecast Months Only	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

※Filter機能では異なるモジュールを選べます

Filter

P2 Products ▼ Time G3 Location Line Items

Show items that match **all** of the following

SYS00 Time Settings: Forecast Period? is equal to

NG Inputモジュールのなかでフィルター条件定義

REV02 Volume Inputs New York

		Apr 20	May 20	Jun 20	Jul 20	Aug 20	Sep 20	Oct 20	Nov 20	Dec 20
Nutzo Bar	Volumes	328	346	389	333	311	272	346	386	354
	Forecast Period?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Raising the Bar	Volumes	342	281	350	347	381	300	311	300	339
	Forecast Period?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Cookie Crumbs	Volumes	333	302	341	342	346	293	260	267	322
	Forecast Period?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Do-nutty Buncha Munch	Volumes	312	330	373	380	387	270	380	302	380
	Forecast Period?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Nougats-N-Cream	Volumes	328	376	316	374	283	261	363	348	334
	Forecast Period?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Nougataciuous	Volumes	315	385	287	365	308	309	341	373	264
	Forecast Period?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

System Module

ほとんどのモデル構築において作成することになる代表的なシステムモジュールを紹介します。

Time Filter

Time FilterはAnaplan構築においてまず最初に習得すべきテクニックの一つです。1モデル内に必ず一つTime Filterモジュールを作成するものと理解してください。InputやOutputモジュールのフィルター条件となります。

Systemモジュールとしてモデル内で一元管理し、様々なInput/Outputモジュールが使用します。Input/Outputモジュールそれぞれでフィルター条件を作ることはやめましょう。

OK Systemモジュールで一元管理

SYS00 Time Settings

	Jan 20	Feb 20	Mar 20	Apr 20	May 20	Jun 20	Jul 20	Aug 20
First day of Month	2020/01/01	2020/02/01	2020/03/01	2020/04/01	2020/05/01	2020/06/01	2020/07/01	2020/08/01
Last day of Month	2020/01/31	2020/02/29	2020/03/31	2020/04/30	2020/05/31	2020/06/30	2020/07/31	2020/08/31
Current Period?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Current Period Start	2020/10/01	2020/10/01	2020/10/01	2020/10/01	2020/10/01	2020/10/01	2020/10/01	2020/10/01
Actual Period?	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Forecast Period?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Forecast Months Only	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

※Filter機能では異なるモジュールを選べます

Filter

P2 Products ▼ Time G3 Location Line Items

Show items that match **all** of the following

SYS00 Time Settings: Forecast Period? is equal to

NG Inputモジュールのなかでフィルター条件定義

REV02 Volume Inputs New York

		Apr 20	May 20	Jun 20	Jul 20	Aug 20	Sep 20	Oct 20	Nov 20	Dec 20
Nutzo Bar	Volumes	328	346	389	333	311	272	346	386	354
	Forecast Period?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Raising the Bar	Volumes	342	281	350	347	381	300	311	300	339
	Forecast Period?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Cookie Crumbs	Volumes	333	302	341	342	346	293	260	267	322
	Forecast Period?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Do-nutty Buncha Munch	Volumes	312	330	373	380	387	270	380	302	380
	Forecast Period?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Nougats-N-Cream	Volumes	328	376	316	374	283	261	363	348	334
	Forecast Period?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Nougataciuous	Volumes	315	385	287	365	308	309	341	373	264
	Forecast Period?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

System Module

ほとんどのモデル構築において作成することになる代表的なシステムモジュールを紹介します。

Detailsモジュール (リストの属性情報管理モジュール)

リストの属性情報や、CODE、NAMEを保持するモジュールです。Systemモジュールとしてモデル内で一元管理し、様々なInput/Outputモジュールが使用します。Input/Calc/OutputモジュールそれぞれでITEM情報を取得することはやめましょう。

OK Systemモジュールで一元管理

Name	Code	Product Family	Size
Nutzo Bar	P001	Chocolates	100g
Raising the Bar	P002	Chocolates	200g
Cookie Crumbs	P003	Chocolates	300g
Do-nutty Buncha Munch	P004	Chocolates	400g
Nougats-N-Cream	P005	Chocolates	500g
Nougataciou	P006	Chocolates	100g
High noon bar	P007	Chocolates	200g
CooCoolicious	P008	Chocolates	300g
That's a Fine Howdy Dot	P009	Chocolates	400g
Dr. Mocho	P010	Chocolates	500g
Craklin Nikklins	P011	Chocolates	100g
Franutbutter Krisp	P012	Chocolates	200g
Unichunky Krisp	P013	Chocolates	300g
PlanutButter Krisp	P014	Chocolates	400g
Chocolates			
Anapoppers	P015	Sours	500g
Juicy Lucy gummies	P016	Sours	100g
Bite Mel	P017	Sours	200g
LocoMong	P018	Sours	300g
Smacker Packe	P019	Sours	400g
Sweet-n-Swirlie	P020	Sours	500g
Souroly polies	P021	Sours	100g

NG Calcモジュールのなかで属性情報取得

		FY20	FY21	FY22
Nutzo Bar	Unit Price %	10.00%	0.00%	0.00%
	Unit Cost %	-10.00%	0.00%	0.00%
Raising the Bar	Unit Price %	10.00%	0.00%	0.00%
	Unit Cost %	-10.00%	0.00%	0.00%
Cookie Crumbs	Unit Price %	10.00%	0.00%	0.00%
	Unit Cost %	-10.00%	0.00%	0.00%
Do-nutty Buncha Munch	Unit Price %	10.00%	0.00%	0.00%
	Unit Cost %	-10.00%	0.00%	0.00%
Nougats-N-Cream	Unit Price %	10.00%	0.00%	0.00%
	Unit Cost %	-10.00%	0.00%	0.00%
Nougataciou	Unit Price %	10.00%	0.00%	0.00%
	Unit Cost %	-10.00%	0.00%	0.00%

NG リストのプロパティとして属性情報保持

Parent	Code	Size
Nutzo Bar	P001	100g
Raising the Bar	P002	200g
Cookie Crumbs	P003	300g
Do-nutty Buncha Munch	P004	400g
Nougats-N-Cream	P005	500g
Nougataciou	P006	100g
High noon bar	P007	200g
CooCoolicious	P008	300g
That's a Fine Howdy Dot	P009	400g
Dr. Mocho	P010	500g
Craklin Nikklins	P011	100g
Franutbutter Krisp	P012	200g
Unichunky Krisp	P013	300g
PlanutButter Krisp	P014	400g
Chocolates		
All Products	CH	
Anapoppers	P015	500g
Juicy Lucy gummies	P016	100g
Bite Mel	P017	200g
LocoMong	P018	300g
Smacker Packe	P019	400g
Sweet-n-Swirlie	P020	500g
Souroly polies	P021	100g
TinyBams	P022	200g
Sours	P023	500g
Rompers	P024	400g
HyperLemones	P025	500g
AnaFrazzles	P026	100g

System Module

ほとんどのモデル構築において作成することになる代表的なシステムモジュールを紹介します。

DCA用モジュール (Dynamic Cell Access権限機能を利用する場合に使用)

動的に権限設定を施したい場合や、セル単位での権限設定が必要な場合、DCA機能を使用することになります。これもInput/Outputモジュール内に定義するのではなく、Systemモジュールとして管理しましょう。

OK Systemモジュールで管理

The screenshot shows the OK System Module interface for 'AD01 Account Hold/Override' in 'FY20'. The interface includes a toolbar with 'View', 'Edit', 'Format', and 'Data' menus. Below the toolbar, there are tabs for 'Hold Write' and 'NOT Hold Read'. The main area displays a list of accounts on the left and a grid of checkboxes for 'Hold Write', 'Hold Read', 'Override Write', and 'Override Read' on the right. The 'Candyate' account is highlighted in blue, and its 'Hold Write' checkbox is checked.

AD01 Account Hold/Override	FY20	Hold Write	Hold Read	Override Write	Override Read
Candyate		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Candy Storemia		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Recandy Store		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Candy Storevu		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Candy Storetude		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Candy Storezio		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Candy Storezu		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Alicandy Store		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Candy Storeboo		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Candy Storella		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Retrocandy Store		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Leecandy		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Metacandy		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Pericandy		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Centicandy		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Candyous		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Socicandy		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

NG In/OutモジュールのなかにDCAドライバーを定義

The screenshot shows the NG In/Out Module interface for 'TAR04 Account Review' in 'FY20'. The interface includes a toolbar with 'View', 'Edit', 'Format', and 'Data' menus. Below the toolbar, there are tabs for 'Access Driver Read' and 'NOT Access Driver Write'. The main area displays a table with columns for 'Previous Year Actual Revenue', 'Initial Account Sales Targets', 'Hold', 'Override', 'Override Sales Target', 'Held or Revised', 'To Change', 'Final Sales Targets', 'Access Driver Write', and 'Access Driver Read'. The 'Candyate' account is highlighted in blue, and its 'Access Driver Read' checkbox is checked.

TAR04 Account Review	FY20	Previous Year Actual Revenue	Initial Account Sales Targets	Hold	Override	Override Sales Target	Held or Revised	To Change	Final Sales Targets	Access Driver Write	Access Driver Read
Candyate		491,779	584,963	<input type="checkbox"/>	<input type="checkbox"/>		0	584,963	580,827	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Candy Storemia		499,166	593,823	<input checked="" type="checkbox"/>	<input type="checkbox"/>		593,823	0	593,823	<input type="checkbox"/>	<input type="checkbox"/>
Recandy Store		497,406	591,681	<input type="checkbox"/>	<input type="checkbox"/>		0	591,681	587,498	<input type="checkbox"/>	<input type="checkbox"/>
Candy Storevu		495,875	589,782	<input type="checkbox"/>	<input type="checkbox"/>		0	589,782	585,612	<input type="checkbox"/>	<input type="checkbox"/>
Candy Storetude		498,256	592,687	<input type="checkbox"/>	<input checked="" type="checkbox"/>	650,000	650,000	0	650,000	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Candy Storezio		89,583	99,606	<input type="checkbox"/>	<input type="checkbox"/>		0	99,606	98,285	<input type="checkbox"/>	<input type="checkbox"/>
Candy Storezu		72,096	80,189	<input type="checkbox"/>	<input type="checkbox"/>		0	80,189	79,126	<input type="checkbox"/>	<input type="checkbox"/>
Alicandy Store		62,843	69,900	<input type="checkbox"/>	<input type="checkbox"/>		0	69,900	68,973	<input type="checkbox"/>	<input type="checkbox"/>
Candy Storeboo		156,877	120,017	<input checked="" type="checkbox"/>	<input type="checkbox"/>		120,017	0	120,017	<input type="checkbox"/>	<input type="checkbox"/>
Candy Storella		133,952	100,750	<input type="checkbox"/>	<input type="checkbox"/>		0	100,750	96,045	<input type="checkbox"/>	<input type="checkbox"/>
Retrocandy Store		144,995	98,409	<input type="checkbox"/>	<input type="checkbox"/>		0	98,409	92,760	<input type="checkbox"/>	<input type="checkbox"/>
Leecandy		143,673	97,591	<input type="checkbox"/>	<input type="checkbox"/>		0	97,591	91,990	<input type="checkbox"/>	<input type="checkbox"/>
Metacandy		110,679	116,916	<input type="checkbox"/>	<input type="checkbox"/>		0	116,916	106,996	<input type="checkbox"/>	<input type="checkbox"/>
Pericandy		138,123	145,935	<input type="checkbox"/>	<input type="checkbox"/>		0	145,935	133,552	<input type="checkbox"/>	<input type="checkbox"/>
Centicandy		132,875	140,294	<input type="checkbox"/>	<input type="checkbox"/>		0	140,294	128,390	<input type="checkbox"/>	<input type="checkbox"/>
Candyous		149,527	157,669	<input type="checkbox"/>	<input type="checkbox"/>		0	157,669	144,290	<input type="checkbox"/>	<input type="checkbox"/>
Socicandy		157,518	167,108	<input checked="" type="checkbox"/>	<input type="checkbox"/>		167,108	0	167,108	<input type="checkbox"/>	<input type="checkbox"/>
Isocandy		87,666	121,975	<input type="checkbox"/>	<input type="checkbox"/>		0	121,975	117,731	<input type="checkbox"/>	<input type="checkbox"/>
Eicandy		65,838	91,522	<input type="checkbox"/>	<input type="checkbox"/>		0	91,522	88,337	<input type="checkbox"/>	<input type="checkbox"/>
Aquacandy		51,798	72,066	<input type="checkbox"/>	<input type="checkbox"/>		0	72,066	69,559	<input type="checkbox"/>	<input type="checkbox"/>
Candyize		76,725	106,734	<input type="checkbox"/>	<input type="checkbox"/>		0	106,734	103,020	<input type="checkbox"/>	<input type="checkbox"/>
Yakicandy		157,566	172,382	<input type="checkbox"/>	<input type="checkbox"/>		0	172,382	170,942	<input type="checkbox"/>	<input type="checkbox"/>
Cryocandy		153,747	167,906	<input type="checkbox"/>	<input checked="" type="checkbox"/>	180,000	180,000	0	180,000	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Procandy		160,396	175,678	<input type="checkbox"/>	<input type="checkbox"/>		0	175,678	174,211	<input type="checkbox"/>	<input type="checkbox"/>
Avicandy		144,683	167,698	<input type="checkbox"/>	<input type="checkbox"/>		0	167,698	155,270	<input type="checkbox"/>	<input type="checkbox"/>
Skacandy		108,055	126,820	<input checked="" type="checkbox"/>	<input type="checkbox"/>		126,820	0	126,820	<input type="checkbox"/>	<input type="checkbox"/>

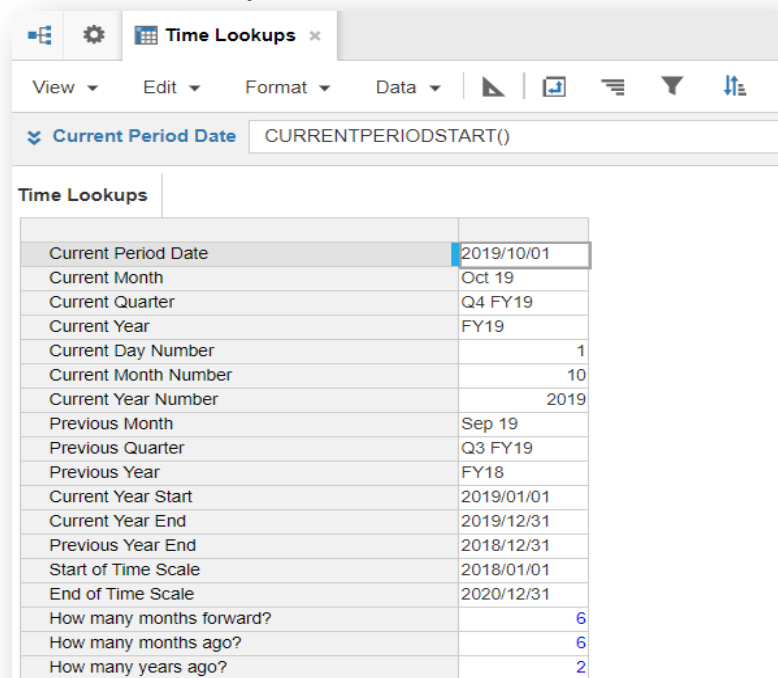
System Module

ほとんどのモデル構築において作成することになる代表的なシステムモジュールを紹介します。

パラメータ値管理モジュール（LookupやIFの条件値を保持するモジュール）

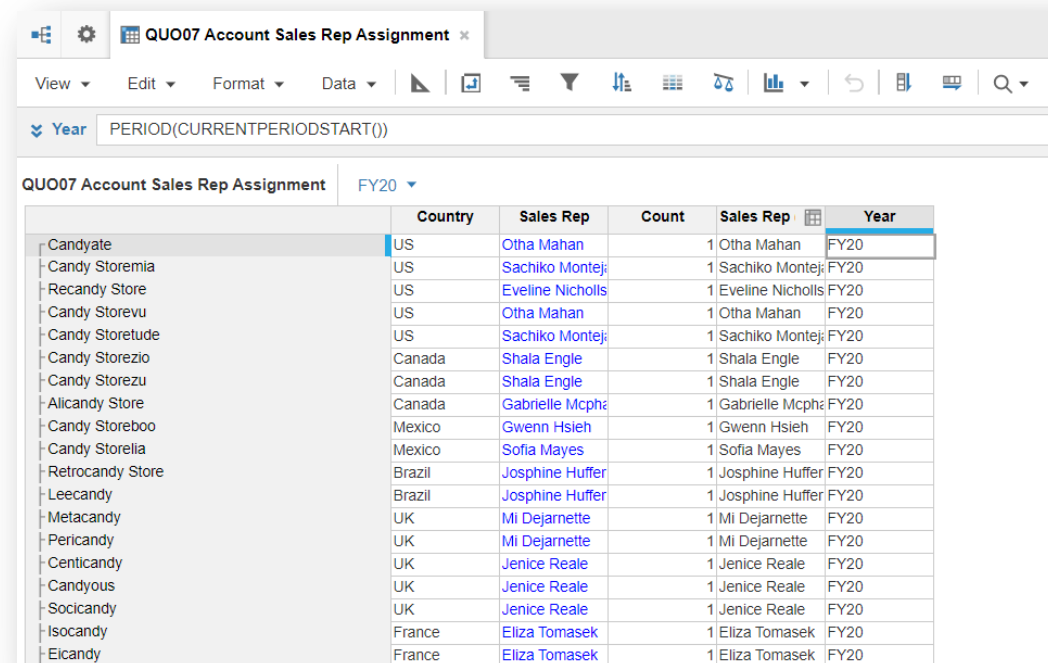
何もDimensionを持たないモジュールにLookupの引数を纏めて管理します。時間/日付系とそれ以外などでSystemモジュールを分けることは問題ありません。

OK Systemモジュールで管理



Time Lookups	
Current Period Date	2019/10/01
Current Month	Oct 19
Current Quarter	Q4 FY19
Current Year	FY19
Current Day Number	1
Current Month Number	10
Current Year Number	2019
Previous Month	Sep 19
Previous Quarter	Q3 FY19
Previous Year	FY18
Current Year Start	2019/01/01
Current Year End	2019/12/31
Previous Year End	2018/12/31
Start of Time Scale	2018/01/01
End of Time Scale	2020/12/31
How many months forward?	6
How many months ago?	6
How many years ago?	2

NG In/Outモジュールのなかで値取得



QUO07 Account Sales Rep Assignment					
	Country	Sales Rep	Count	Sales Rep	Year
Candyate	US	Otha Mahan	1	Otha Mahan	FY20
Candy Storemia	US	Sachiko Monteji	1	Sachiko Monteji	FY20
Recandy Store	US	Eveline Nicholls	1	Eveline Nicholls	FY20
Candy Storevu	US	Otha Mahan	1	Otha Mahan	FY20
Candy Storetude	US	Sachiko Monteji	1	Sachiko Monteji	FY20
Candy Storezio	Canada	Shala Engle	1	Shala Engle	FY20
Candy Storezu	Canada	Shala Engle	1	Shala Engle	FY20
Alicandy Store	Canada	Gabrielle Mcph	1	Gabrielle Mcph	FY20
Candy Storeboo	Mexico	Gwenn Hsieh	1	Gwenn Hsieh	FY20
Candy Storelia	Mexico	Sofia Mayes	1	Sofia Mayes	FY20
Retrocandy Store	Brazil	Josphine Huffer	1	Josphine Huffer	FY20
Leecandy	Brazil	Josphine Huffer	1	Josphine Huffer	FY20
Metacandy	UK	Mi Dejarnette	1	Mi Dejarnette	FY20
Pericandy	UK	Mi Dejarnette	1	Mi Dejarnette	FY20
Centicandy	UK	Jenice Reale	1	Jenice Reale	FY20
Candyous	UK	Jenice Reale	1	Jenice Reale	FY20
Socicandy	UK	Jenice Reale	1	Jenice Reale	FY20
Isocandy	France	Eliza Tomasek	1	Eliza Tomasek	FY20
Eicandy	France	Eliza Tomasek	1	Eliza Tomasek	FY20

Calculate Once, Reference Many Times

ここまで紹介してきたSystemモジュールの役割は、制御系の値を一箇所で計算・管理し、In、Calc、Outモジュールそれぞれから参照されるためです。「Calculate Once, Reference Many Times」のルールを必ず守りましょう。同じFormulaを何度も書いていたら何かおかしいと立ち止まって考えてください。

START() <= CURRENTPERIODSTART()が5回、START() > CURRENTPERIODSTART()が2回繰り返されている

Time以外にも次元が沢山あるため、必要以上に無駄な計算が多く走っている状態

NG

	Formula	Parent	Is Summary	Format	Applies To	Time Sci
Call Data Forecast					Organization, Hour Scale, Call Center Type	Day
Override Call Flag	IF START() <= CURRENTPERIODSTART() THEN 0 ELSE IF Call Center Forecast Override		<input type="checkbox"/>	Number		Day
Override Avg Handle Time Flag	IF START() <= CURRENTPERIODSTART() THEN 0 ELSE IF Call Center Forecast Override		<input type="checkbox"/>	Number		Day
Distance from Actuals	Current Period Actuals Through - START() + 1		<input type="checkbox"/>	Number		Day
Total Actual Calls	IF START() > CURRENTPERIODSTART() THEN 0 ELSE Import Call Data.Inbound.Y		<input type="checkbox"/>	Number		Day
1_Max_Calls			<input type="checkbox"/>	Number		Day
X_All_Calls	CUMULATE(1)		<input type="checkbox"/>	Number		Day
X_Max_Calls	IF Y_Calls < 0 THEN CUMULATE('1_Max_Calls') ELSE 0	X_Total_Calls	<input type="checkbox"/>	Number		Day
X_Total_Calls	X_Max_Calls		<input checked="" type="checkbox"/>	Number		Day
X_Sqr_Calls	POWER(X_Max_Calls, 2)		<input type="checkbox"/>	Number		Day
Y_Calls	Total Actual Calls		<input checked="" type="checkbox"/>	Number		Day
m_Calls	X_Max_Calls * Y_Calls		<input type="checkbox"/>	Number		Day
c_Calls	(Y_Calls[SELECT: TIME.All.Periods] - m_Calls * X_Total_Calls[SELECT: TIME.All.Periods]) / X_Max_Calls		<input type="checkbox"/>	Number		Day
E(Y) - Straight Line Calls			<input type="checkbox"/>	Number		Day
E(Y) - Moving Average Calls	IF START() <= CURRENTPERIODSTART() THEN Y_Calls ELSE MOVINGSUM()		<input type="checkbox"/>	Number		Day
E(Y) - Exponential Smoothing Calls Formula	IF CUMULATE(1) = 1 THEN Y_Calls ELSE Trend Line Method.alpha for Exponential Smoothing		<input type="checkbox"/>	Number		Day
E(Y) - Exponential Smoothing Calls	IF START() <= CURRENTPERIODSTART() + 1 AND END() > CURRENTPERIODSTART() THEN Y_Calls ELSE MOVINGSUM()		<input type="checkbox"/>	Number		Day
Total Inbound	IF START() <= CURRENTPERIODSTART() THEN Total Actual Calls ELSE IF START() > CURRENTPERIODSTART() THEN 0 ELSE Import Call Data.Average.Avg Handle Time		<input type="checkbox"/>	No Data		Day
Total Actual Avg Handle Time	IF START() > CURRENTPERIODSTART() THEN 0 ELSE Import Call Data.Average.Avg Handle Time		<input type="checkbox"/>	Number		Day
1_Max_Avg Handle Time	IF 1_Max_Avg Handle Time < 0 THEN 1_Max_Avg Handle Time ELSE 0		<input type="checkbox"/>	Number		Day
X_All_Avg Handle Time	(1_Max_Avg Handle Time * 1_Max_Avg Handle Time) / 1_Max_Avg Handle Time		<input type="checkbox"/>	Number		Day

TimeのみをDimensionに持つSystemモジュールで計算

OK

	Formula	Parent	Is Summary	Format	Applies To	Time
Days of Week						Day
Weekday or Weekend	IF WEEKDAY(START()) = 6 OR WEEKDAY(START()) = 7 THEN Weekday?.Weekend		<input type="checkbox"/>	Weekday?		Day
Day of Week	IF WEEKDAY(START()) = 1 THEN Monday ELSE IF WEEKDAY(START()) = 2 THEN Tuesday ELSE IF WEEKDAY(START()) = 3 THEN Wednesday ELSE IF WEEKDAY(START()) = 4 THEN Thursday ELSE IF WEEKDAY(START()) = 5 THEN Friday ELSE IF WEEKDAY(START()) = 6 THEN Saturday ELSE IF WEEKDAY(START()) = 7 THEN Sunday		<input type="checkbox"/>	Day of Week		Day
History Period?	START() <= CURRENTPERIODSTART()		<input type="checkbox"/>	Boolean		Day
Future Period?	CURRENTPERIODSTART() < START()		<input type="checkbox"/>	Boolean		Day





Systemモジュールを参照する

	Formula	Parent	Is Summary	Format	Applies To	Time
Call Data Forecast						
Override Call Flag	IF Days of Week.History Period? THEN 0 ELSE IF Call Center Forecast Override		<input type="checkbox"/>	Number		Day
Override Avg Handle Time Flag	IF Days of Week.History Period? THEN 0 ELSE IF Call Center Forecast Override		<input type="checkbox"/>	Number		Day
Distance from Actuals	Current Period Actuals Through - START() + 1		<input type="checkbox"/>	Number		Day

Subsidiary Viewは必要以上に使わないこと

モジュール作成時に、そのモジュールに適用するリストを指定しますが、その後Line Itemごとに適用するリストを増やしたり減らしたりすることができます。モジュールに対して適用されているリストと、Line Itemのそれが異なる場合、Line Itemに対する適用リストを確認するための、「Subsidiary View」と呼ばれるビューが出現します。

Subsidiary Viewが使われていると、第三者がロジックを見た際に、理解することが非常に困難になるため、Calcモジュールに使うことは推奨されません。適用するリストが異なるのであれば、それはモジュールから分けましょう。

Account & Employees Module					GL Accounts, Employees	
Yearly Revenue			<input type="checkbox"/>	Number	-	
Yearly Costs			<input type="checkbox"/>	Number	-	
Yearly Margin			<input type="checkbox"/>	Number	-	
Country		Employees.Country List		<input type="checkbox"/>	Country	Employees
Sub Region		Employees.Sub Region		<input type="checkbox"/>	Text	Employees
Region		Employees.Region		<input type="checkbox"/>	Text	Employees
Area		Employees.Area		<input type="checkbox"/>	Text	Employees

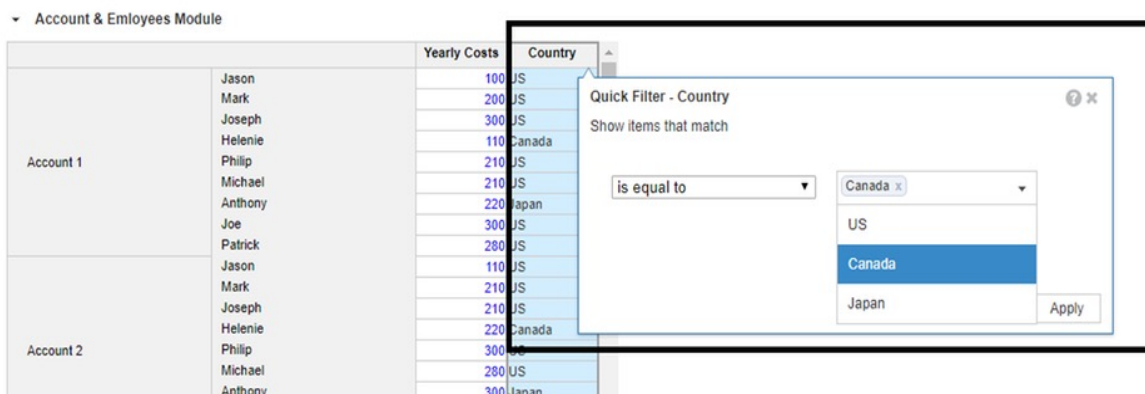
この小さいマークがSubsidiary Viewでカーソルを合わせてクリックするとLine Item単位のビューに遷移する

ハイフンは、モジュールに対して設定している Applies Toと同じリストを適用するという意味の設定。

Subsidiary Viewの活用場面

一方で、「Subsidiary View」を使用する方が適切な例もあります。そのため、絶対に使ってはならないというわけではなく、明確な意図のもと使うことは問題ないご理解ください。(モジュールの切り分けが面倒だからという理由や、知らないうちにSubsidiaryが発生していたという状況がNG)

Output画面上での属性情報表示やフィルター、ソート条件にする場合



Account & Employees Module

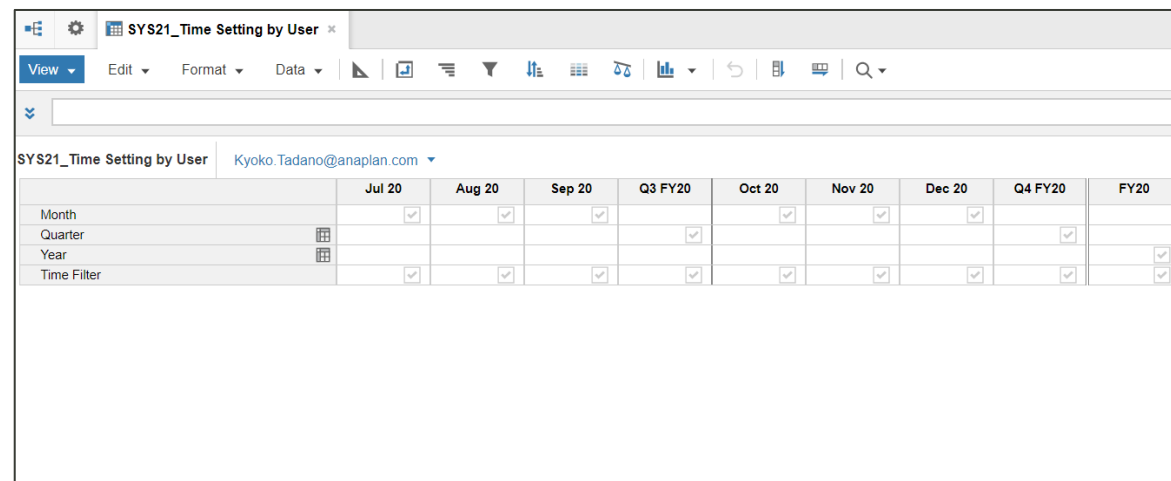
Account	Employee	Yearly Costs	Country
Account 1	Jason	100	US
	Mark	200	US
	Joseph	300	US
	Helenie	110	Canada
	Philip	210	US
	Michael	210	US
	Anthony	220	Japan
	Joe	300	US
	Patrick	280	US
	Jason	110	US
Account 2	Mark	210	US
	Joseph	210	US
	Helenie	220	Canada
	Philip	300	US
	Michael	280	US
	Anthony	300	Japan

Quick Filter - Country
Show items that match
is equal to
Canada x
US
Canada
Japan
Apply

アウトプットモジュール内に表示すべき情報で、一部の軸にのみ依存する情報を表示するケースを想定。

例えば、上記はアカウントリストと従業員リストの2軸が適用されたモジュールです。ユーザーの参考情報として同一モジュール上に従業員の国を表示する追加の列を持ちたいとしたとき、その情報は従業員リストにのみ依存する情報なので、アカウントリストがあると無駄なセルが発生してしまうこととなります。このような場合は、当該ラインアイテムからアカウントリストを除き従業員リストのみが適用されたラインアイテムで国情報を取得します。

CalcやSysモジュールであっても、複数モジュールに分けずに一纏めにした方が管理し易い場合



SYS21_Time Setting by User

	Jul 20	Aug 20	Sep 20	Q3 FY20	Oct 20	Nov 20	Dec 20	Q4 FY20	FY20
Month	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Quarter				<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>	
Year									<input checked="" type="checkbox"/>
Time Filter	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

上記は上からそれぞれMonth/Quarter/YearとTime Scaleが異なるラインアイテム。

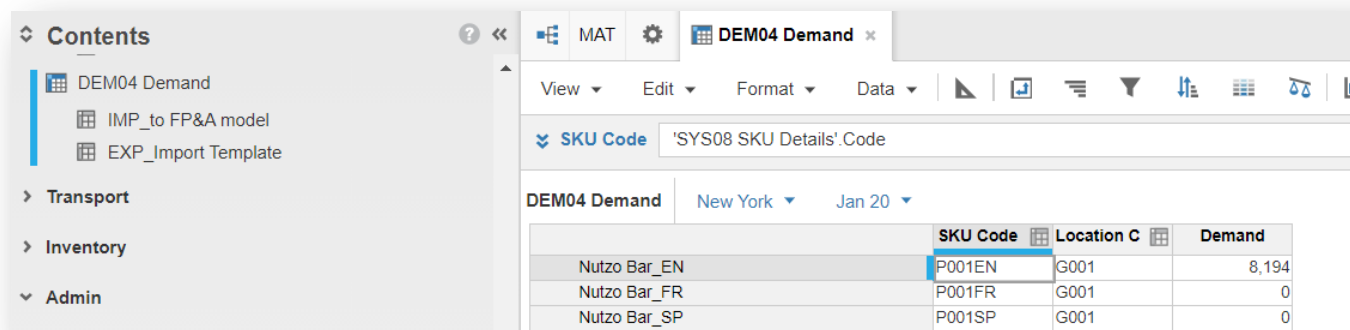
何が選択されているか同一モジュール内で分かる方が管理しやすいため Subsidiary Viewで定義している例。

Saved Viewの用途

作成したモジュールに対してSaved Viewというビューを定義することができます。Saved Viewはとても重要な役割を果たします。用途をきちんと理解して、適切に使いましょう。

※クラシックダッシュボードに馴染みがある人は、UI用にSaved Viewを使ってしまいがちですが、New UXでは、Saved View非推奨です。注意しましょう。

1. Importのソースデータ： Anaplan内でImportをする場合、ソースをList、Module、Saved Viewの3種類から選べるが、**必ずSaved Viewを作成し、それをソースとすること。**
2. Exportのソースデータ： どのようなレイアウトやフィルター条件でExport定義が作成されたか、後から確認ができるように、**必ずSaved Viewを作成し、それをソースとすること。**
3. その他： モデルビルダーのデータバリデーション用など



The screenshot shows the Anaplan interface with a table titled 'DEM04 Demand'. The table has columns for 'SKU Code', 'Location C', and 'Demand'. The data is filtered by 'SYS08 SKU Details'. Code' and 'New York' location, with a date of 'Jan 20'. The table contains three rows of data.

SKU Code	Location C	Demand
Nutzo Bar_EN	P001EN	8,194
Nutzo Bar_FR	P001FR	0
Nutzo Bar_SP	P001SP	0

データの絞り込みについて

Anaplanではデータの絞り込み機能は3つに分けられます。Page Selector(標準)、フィルター機能(標準)、Line Itemを活用したフィルター(カスタム)です。それぞれの特徴を理解しましょう。

1、Page Selector (コンテキストセクター)

- リストの要素を指定すると、即時でSyncされる。階層リストの際に強みを発揮する。
- 同じ軸であれば異なるモジュール間でSyncされる。Sync有無はカード単位で選択可能。
- ユーザーごとの固有情報として扱われる。Resetしない限り、最後に選んだ状態が保たれる(NUXの場合)

NUX



Classic



2、標準のフィルター機能

- セルの値を条件にフィルターをかける
- 使いこなせるまで慣れが必要 (多軸の場合特に)
- 複数条件の掛け合せ等、複雑なフィルタリングはできない

NUX



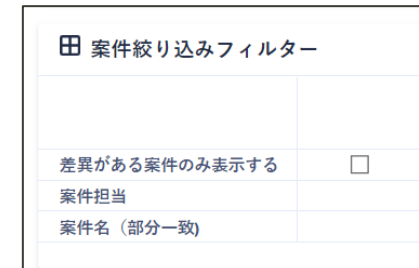
Classic



3、Line Itemを活用したフィルター

- フィルター条件を入力する箱をラインアイテムで定義し、その値をもとに該当データをBooleanで判定、Boolean値を標準のフィルター条件に設定する方法
- 次ページにて詳細説明

NUX



Classic



Line Itemを活用したフィルター (1/2)

ユーザーにBoolean値で選択させるパターン

	US	Canada	Mexico	Brazil	UK	France	Spain	Germany	Italy	Australia	New Zealand	India	Japan
Chocolates	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Sours	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Taffy	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Fudge	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

		FY20	FY21	FY22
Chocolates	Revenue	452,784	456,677	0
Chocolates	Cost of Sales	126,017	127,071	0
Chocolates	Margin	326,766	329,606	0
Chocolates	Margin %	72.17%	72.17%	0.00%
Sours	Revenue	533,946	541,067	0
Sours	Cost of Sales	137,166	138,975	0
Sours	Margin	396,780	402,092	0
Sours	Margin %	74.31%	74.31%	0.00%

Page側でフィルター対象モジュールに条件をセット

フィルター

▼ 行 列

P1 Product Family > Line Items

次と一致するアイテムを表示:

Rule

SYS10_Filter Test Pattern1

Show?

等しい

コンテキスト: セレクター (- 現在 -)

ルールの追加

すべてのフィルターを解除

ユーザーにドロップダウンから選択させるパターン

Dropdown

Region: Europe

この日付以降に入社: 2016/05/01

	Code	Name	Department	Country	Region	Start Date	Leave Date	Current Salary	Commission %
Mi Dejarnette	043	Mi Dejarnette	Sales	UK	Europe	2016/05/01		36,000	10.00%
Jenice Reale	049	Jenice Reale	Sales	UK	Europe	2016/09/01		79,000	10.00%
Sharilyn Lubinsky	052	Sharilyn Lubinsky	Sales	France	Europe	2017/02/01		70,000	10.00%
Trena Herald	069	Trena Herald	Sales	Italy	Europe	2018/03/01		38,000	10.00%
Otilia Burke	071	Otilia Burke	Sales	Spain	Europe	2018/05/01		63,000	10.00%
Cami Haire	081	Cami Haire	Sales	Spain	Europe	2019/03/01		51,000	9.00%
Wally Hartnett	100	Wally Hartnett	Sales	Italy	Europe	2020/08/01		27,000	9.00%
Angela Billings	110	Angela Billings	Sales	Germany	Europe	2021/05/01		71,000	8.00%
Alvaro Defrancisco	053	Alvaro Defrancisco	Production	UK	Europe	2017/02/01		51,000	0.00%
Vashti Ellingson	054	Vashti Ellingson	Production	Italy	Europe	2017/03/01		84,000	0.00%
Marcia Crapo	058	Marcia Crapo	Production	Spain	Europe	2017/08/01		98,000	0.00%
Shantae Wilton	067	Shantae Wilton	Production	Spain	Europe	2018/01/01		60,000	0.00%

選択値を基にロジックでBoolean値を作成

SYS12_Filter Test Pattern2

View Edit Format Data

Show

(ISBLANK(SYS11_Filter Test Pattern2: Region) OR SYS11_Filter Test Pattern2: Region = SYS08_Employee Details: Region) AND (ISBLANK(SYS11_Filter Test Pattern2: この日付以降に入社) OR SYS11_Filter Test Pattern2: この日付以降に入社 <= SYS08_Employee Details: Start Date)

Shala Engle
Otha Mahan
Sindy Wager
Eikza Tomasek
Katalin Scarpelli
Sommer Teasley
Josgine Huffer
Theda Dickerson
Lashonda Dmkard
Brandon Welker
Mi Dejarnette
Janell Wetzell
Jenice Reale
Myrie Capshaw
Sharilyn Lubinsky
Gabrielle Mcphatter
Sachiko Montejano
Trena Herald
Otilia Burke
Mckenzie Rozell
Cami Haire
Felix Tate

ルールの追加

すべてのフィルターを解除

Page側でフィルター対象モジュールに条件をセット

フィルター

▼ 行 列

E2 Employees#

次と一致するアイテムを表示:

Rule

SYS12_Filter Test Pattern2

Show

等しい

ルールの追加

すべてのフィルターを解除

Line Itemを活用したフィルター (2/2) Users軸有無の違い

Line Itemのフィルターは、ユーザー間で共通の条件になります。管理者が設定するようなフィルターであれば問題ありませんが、ユーザー個々のフィルター条件になるべきであれば、Users軸を追加します。

The image shows a sequence of three screenshots illustrating the configuration of a filter in Anaplan. The first screenshot shows the 'Region' filter set to 'Americas' and the 'Users' dropdown set to 'Kyoko.Tadano@anaplan.com'. The second screenshot shows the resulting data table with columns for 'Users' and rows for various employees, with the first row highlighted. The third screenshot shows the 'Filter' dialog box with the 'Users' context selector set to 'Current User'.

Filter Configuration:

- Filter Name: SYS12_Filter Test Pattern2
- Rule: (ISBLANK('SYS11_Filter Test Pattern2'.Region) OR 'SYS11_Filter Test Pattern2'.Region = 'SYS08 E
- Context Selector: Users
- Selected User: Current User

Users	Shala Engle	Otha Mahan	Sindy Wager	Eliza Tomasek	Katelin Scarpelli	Sommer Teasley	Josphine Huffer	Theda Dickerson	Lashonda Drinkard	Branden Welker	Mi Dejarnette	Janell Wetzel	Jenice Reale	Myrle Capshaw	Sharilyn Lubinsky	Gabrielle Mcphatter	Sachiko Montejano	Trena Herald	Otilia Burke	Mckenzie Rozell	Cami Haire	Felix Tate	Sofia Mayes	Eveline Nicholls	Lahoma Sabine
Kyoko.Tadano@anaplan.com																									

Usersの要素は、そのモデルにアクセス権を持つ（No Access以外）ユーザー一覧となります。全てのUserが表示されるのはモデルビルダーのみで、エンドユーザーは自分のユーザーIDのみが見えます。※他のユーザーの条件を変えてしまうという事は発生しない。

フィルターのかけ方

フィルターをかける場合、その条件となるLine ItemのFormatはBoolean型にしましょう。AnaplanではBoolean型が1番パフォーマンス面で優れています。容量の消化も1セル1byteです。(Number型は8byte)

例えば、フィルターの条件に、直接「Value > 100」などの条件を設定するのではなく、Boolean FormatのLine ItemでValue > 100の判定を済ませておいて、そのTRUE, FALSEをFilterの条件します。そうすることで、UI表示のパフォーマンスが向上します。

また複数条件が伴う場合、複数のBoolean型ラインアイテムを直接フィルタに設定するのではなく、その複数条件を判定する最終のLine Itemを一つ追加し、その一つのみをフィルター条件に設定するようにします。

OK 複数の条件判定結果を最後に一つ別のラインアイテムでまとめてそれをフィルター条件として設定する。

S_FTR_案件/見込Version		
FLG_差異がある案件	Boolean	IF NOT S_FTR_Select by User: 差異がある案件のみ表示する THEN TRUE ELSE FALSE
FLG_担当	Boolean	ISBLANK(S_FTR_Select by User: 案件担当) OR S_FTR_Select by User: 案件担当
FLG_案件名	Boolean	IF ISBLANK(S_FTR_Select by User: 案件名 (部分一致)) THEN TRUE ELSE FALSE
<u>FLG_ALL</u>	Boolean	FLG_差異がある案件 AND FLG_担当 AND FLG_案件名

The screenshot shows the 'Filter' configuration window. The 'Show items that match' dropdown is set to 'all'. A single filter condition is visible: 'S_FTR_案件/見込Version: FLG_ALL' with the operator 'is equal to' and a checked checkbox. The 'Users' dropdown is set to '-- Current User --'.

NG 各フィルター条件判定結果をフィルター機能上で直接複数設定してしまう

The screenshot shows the 'Filter' configuration window with three filter conditions stacked vertically. Each condition has the operator 'is equal to' and a checked checkbox. The conditions are: 'S_FTR_案件/見込Version: FLG_差異がある案件', 'S_FTR_案件/見込Version: FLG_担当', and 'S_FTR_案件/見込Version: FLG_案件名'. The 'Users' dropdown is set to '-- Current User --'.

OutputがUsers依存の場合、容量に注意

フィルターだけではなく、アウトプットの値自体を、Users固有の条件によって変えたい、という要件の場合は注意が必要です。例えば、ユーザーが選択した製品のみの上合計値を表示したい、などです。

この場合フィルターするだけではなく、値自体をUser単位で保持する必要があるため、アウトプットモジュール側にもUsers軸をつける必要があります。少ないユーザー数であれば問題ありませんが、数百~になるとその分セルが確保されるので容量が多く消費されます。

特に危険なのが、このことを理解せずに構築を進めると、開発環境は通常ユーザー数が少ないため問題が検知できず、本番モデルを立ててユーザー登録をした瞬間に容量が爆発するという自体が発生します。Users軸を付ける場合は、本番相当の件数を容量試算の対象に含めることを忘れないようにしましょう。

ユーザー条件モジュール

	US
Chocolates	<input checked="" type="checkbox"/>
Sours	<input type="checkbox"/>
Taffy	<input checked="" type="checkbox"/>
Fudge	<input checked="" type="checkbox"/>
All Products	<input checked="" type="checkbox"/>

アウトプットモジュール: ユーザー条件を参照して、対象であれば値取得、対象外はゼロを取得

	FY20	FY21	FY22	All Periods
Chocolates	301,216	242,614	0	543,830
Sours	0	0	0	0
Taffy	283,144	214,883	0	498,028
Fudge	162,766	120,719	0	283,485
All Products	747,127	578,216	0	1,325,343

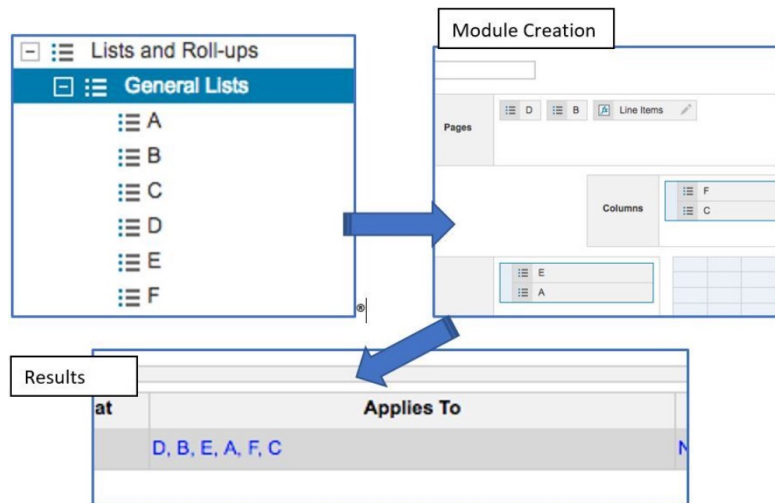
※ユーザー数が多くアウトプットをUsers付きで保持することは容量の観点で不可能となる場合、AnaplanのExcel Addinを利用する、BIツールへ機能棲み分けをするなどAnaplan UI以外で対応する方法が現状考えられるワークアラウンドです。

Applies Toの順番

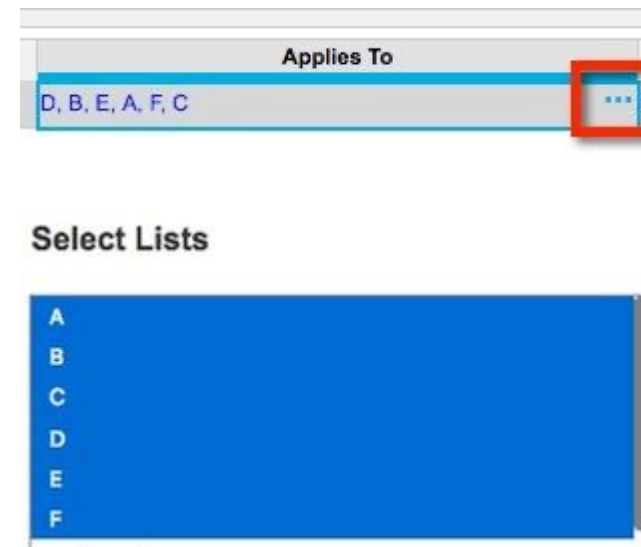
モジュールのApplies Toの順番はモジュール間で揃えるようにしましょう。揃っていないとパフォーマンスに悪影響を与えます。

- モジュール作成時のGeneral Listの順番でApplies Toは作られます。General Listの順番を変更しても、Applies Toは作成時の順番のままです。
- Applies Toの三点リーダーから「ダイアログを開いてOK」をすることで、各モジュールのApplies Toの順番は、General Listの順番も揃います。

モジュール作成時



3点リーダーから直す



OK!



Applies Toの順序がパフォーマンスに関係する理由は、はApplies ToにあるDimensionの順でMappingが行われるためです。

技術的な詳細を知りたい方は以下のリンクから。

<https://community.anaplan.com/t5/Knowledge/Dimension-Order/ta-p/32934>

モデルビルダーのための10か条



<https://community.anaplan.com/t5/Preface/tkb-p/Chapter0Planual>

黒字は頭の片隅に留めておいていただきたいガイドラインです。
青字は本資料内で具体的な開発テクニックとして触れます。

1. 力技の作り込みは必ずしもやるべきとは限らない
(Just because you CAN doesn't mean you SHOULD)
2. 計算は1度、参照を複数回
(Calculate once, reference many time)
3. 常にシンプルな設計を心掛ける
(Simple > Complex)
4. 必要な箇所だけに計算させる
(Only calculate when you need to)
5. スパース性は必ずしも悪いとは限らない
(Sparsity is not your enemy)
6. (モデルサイズが)大きくパフォーマンスに優れたモデルは、小さく遅いモデルより良い
(Big and fast > small and slow)
7. Boolean型 = 軽い テキスト型 = 重い
(Booleans = great. Text = bad)
8. テキスト結合処理 = さらに重い
(Text joins = worse)
9. イレギュラーケースによってモデルを複雑化させることを避けること
(Don't let exceptions overcomplicate the model)
10. SUMとLOOKUPの併用は必ず避けること
(Sums and lookups are good, but never together)

Thank you!