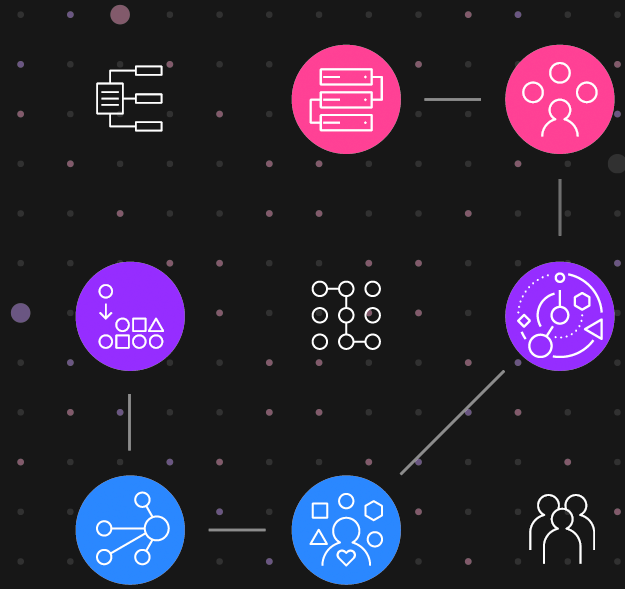




Deep dive Cognos Analytics Data Modules

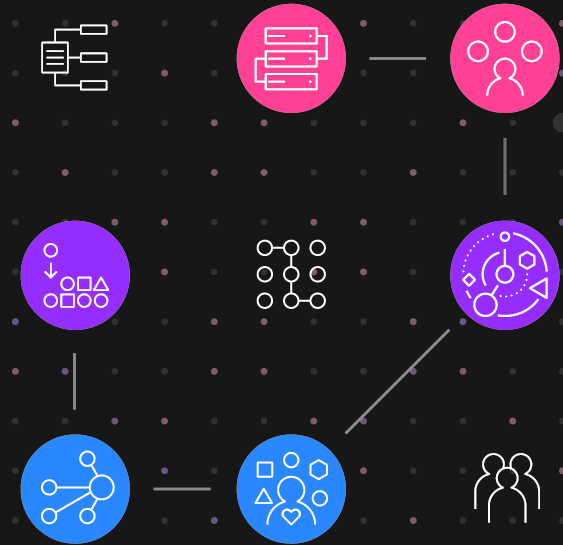
—
Oct 7, 2020



Shaping the future of smart data preparation in IBM Cognos Analytics

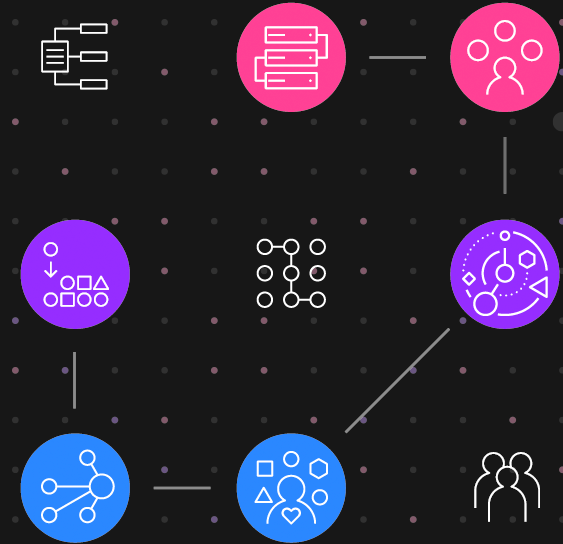
—
Torben Noer
Offering Manager Analytics
Torben.noer@dk.ibm.com

Peter D'Haeyer
Analytics SaaS Technical Lead - Emea
Peter.dhaeyer@be.ibm.com



Agenda

- FM vs Data Modules
- Overview
- Tips & Tricks & Work arounds



**Framework
Manager**

vs

**Data
Modules**

The Basics

- ✓ Flat file – Multi tab and append
- ✓ Smart relationships
- ✓ Data Prep – Split column, trim,... Hierarchies, data groupings
- ✓ Relative Times
- ✓ Aggregation across grains
- ✓ Ease of use – expression editor, SQL based tables, folders, format, filter
- ✓ Set Operations – Union, Intersect, Except
- ✓ Security Filters

Key New Capabilities *Modules*

- Graduated experience – business analysts to experienced users
- Easy to use, yet powerful when needed
- Almost at par with Framework Manager

Use a data module unless you can't !

- Easy and fast to create
- Reusable and combinable
- Web based
- Has capabilities not available in FM
- Investments will happen in DM
- Gaps with FM are narrowing!
- Architected to be the best of both worlds
- Simplicity and performance of relational
- Functionality of dimensional

New in 11.1.7

Centralized place in standard documentation for

FM vs DM gaps & differences:

https://www.ibm.com/support/knowledgecenter/en/SSEP7J_11.1.0/com.ibm.swg.ba.cognos.ca_mdmg.doc/c_bp_dm_diffs.html

Modeling best practices :

https://www.ibm.com/support/knowledgecenter/SSEP7J_11.1.0/com.ibm.swg.ba.cognos.mod_guidelines.doc/c_mod_guidelines.html

Please read the full article here (Cognos Analytics User Community):

<https://community.ibm.com/community/user/businessanalytics/blogs/torben-noer1/2020/09/08/metadata-modeling?CommunityKey=6b10df83-0b3c-4f92-8b1f-1fd80d0e7e58&tab=recentcommunityblogsdashboard>

Where have we been?



Evolution since 11.1.4

Improved joins

- Support for 'between'
- Non-equi joins operators: [$<$, $>$, $<=$, $>=$] now supported

Match selected columns

| Manager Code 4 | Manager Mb 4 | Manager Code 5 | record start date key ORDER_DAY_KEY | Record end date key OrdDate | DateStr | Day | mon |
|----------------|--------------|----------------|--|--------------------------------|------------|-----|-----|
| 10576 | 井森 源藏 | 10087 | 20090109 | 20190627 | 2010-01-12 | 12 | 0: |
| 10576 | 井森 源藏 | 10087 | 20070912 | | | | 0: |
| 10652 | 田中 三部 | 10551 | 20080613 | | | | 0: |
| 10366 | Dale Thomas | 10366 | 20030812 | | | | 0: |

Defined matches 1

| | | | |
|-----------------------|--------|---------------|---|
| record start date key | \leq | ORDER_DAY_KEY | - |
| Record end date key | \geq | OrdDate | - |

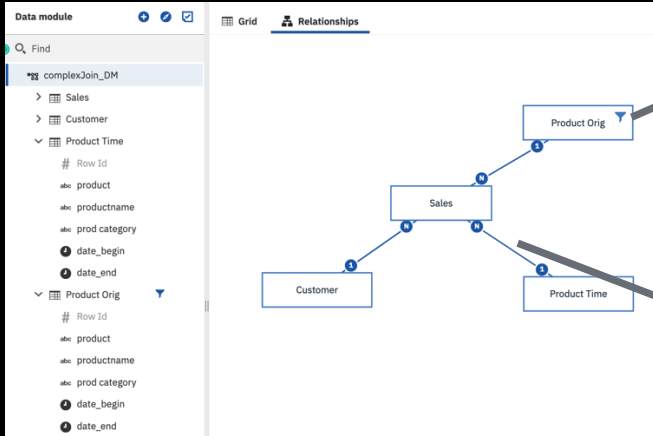
2 Matched columns

Inner join, 1-to-many, No filtering.

OK Cancel

Complex joins

- Allows to model time depending dimensions and views
- Use filters to limit to one version



Edit filter

Name: C_date_begin

Components: Product Orig

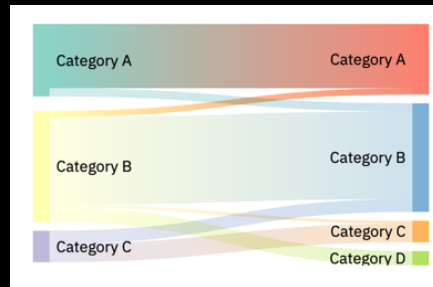
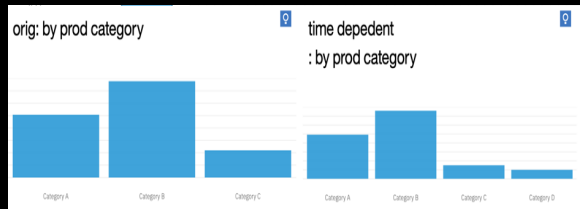
Expression: 1 | year(date_begin)='2017'

Match selected columns

| Table 1 | Table 2 |
|------------|---------------|
| Row Id | product |
| Date | productname |
| Product | prod category |
| date_begin | date_begin |
| date_end | date_end |

Defined matches

| Table 1 | Table 2 | Matched columns |
|---------|------------|-----------------|
| Product | product | 1 |
| Date | date_begin | 2 |
| Date | date_end | 3 |



Data module

| Row Id | product | productname | prod category | date_begin | date_end |
|--------|---------|-------------|---------------|------------|----------|
| 1 | P1 | Product 1 | Category A | 1/1/17 | 12/31/18 |
| 7 | P1 | Product 1 | Category A | 1/1/19 | 12/31/23 |
| 2 | P2 | Product 2 | Category B | 1/1/17 | 12/31/18 |
| 8 | P2 | Product 2 | Category A | 1/1/19 | 12/31/23 |

Simplified data group creation

- Smart data groups proposed (AI)
- Change from numeric to text style
- In cases where groups need to have specific values grouped together



Create a data group (numeric style)

Name Age (Group) (3)

| Group names | Groups | 5 | Range border values |
|--------------|--------|---|---------------------|
| 52 and above | | | Higher |
| | | | 52 |
| 44 to < 52 | | | 44 |
| | | | 35 |
| 35 to < 44 | | | 27 |
| | | | Lower |
| 27 to < 35 | | | |
| less than 27 | | | |

Group NULL values as

Create a data group (text style) Create Cancel

Create a data group (text style)

Name Age (Group) (3)

Remaining items in column

Find

18
19
20
21
22
23

Clear 0 of 43 selected Invert

Groups

+ New group

0 of 0 selected

Group items

Select a single group to view its items

Clear 0 of 0 selected Invert

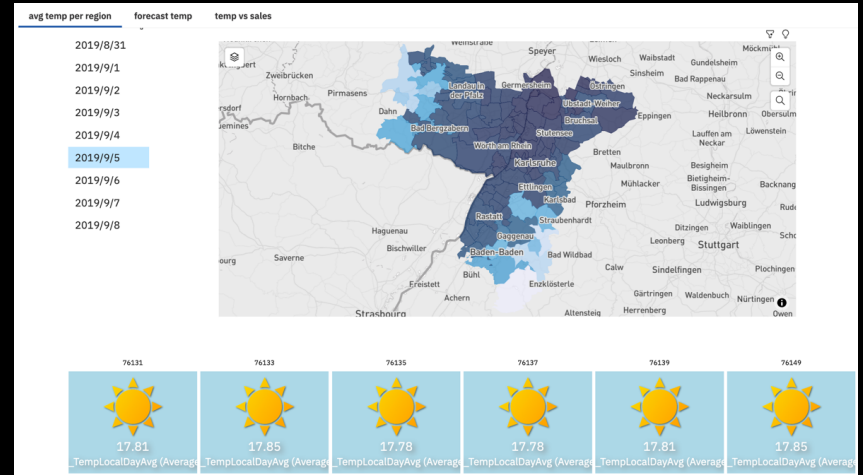
Group remaining and future items in

Replace the existing column 'Age'

Create a data group (numeric style) Create Cancel

IBM Weather Company Data

- Live data connectivity to IBM Weather Company's **History on Demand** and **Enhanced Forecast** API packages
- Blend weather data with your corporate data

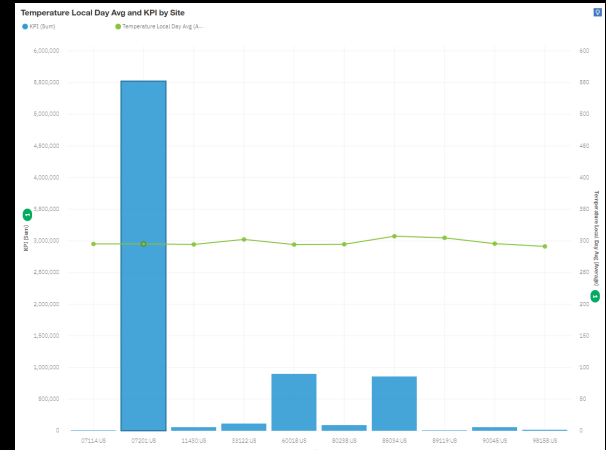


The screenshot shows the IBM Cognos Analytics interface. On the left, there is a sidebar with navigation icons. The main area is titled "Data server connections" and contains a table of connections. On the right, there is a "Select a type" dropdown menu with a search bar and a list of data sources. A blue arrow points to the "IBM Weather Company" entry in the list.

| Name | Modified |
|-------------------------|--------------------|
| Audit | 7/11/2018 10:35 PM |
| BRS - Comercial | 3/14/2019 10:35 AM |
| Bluemix DB2 GoSales -RO | 7/31/2018 1:43 PM |
| Credit Risk | 9/13/2018 3:06 PM |
| DASHDB_STP | 8/9/2018 9:04 PM |
| DB2WC_MF | 10/2/2018 9:03 AM |

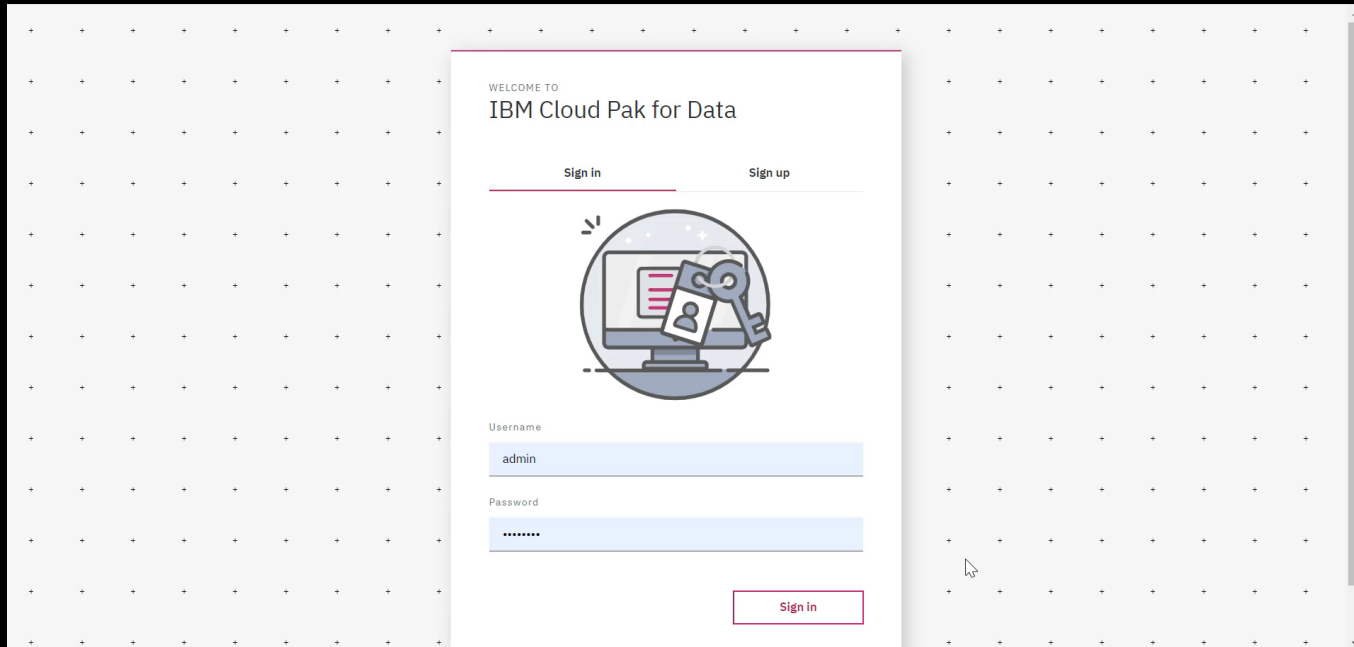
Select a type

- IBM DB2 warehouse
- IBM Informix Dynamic Server
- IBM Netezza
- IBM Planning Analytics
- IBM Weather Company**
- MariaDB
- Microsoft Analysis Services
- Microsoft SQL Server
- MongoDB Connector for BI
- MySQL



Watson Knowledge Catalog integration – phase 1

One source of definition/reuse of data server connections defined in Watson Knowledge Catalog



REST API datasources

New way to connect to a multiple range of datasources that have documented API

The screenshot shows the IEXTRADING application interface. On the left, there's a sidebar with 'Owner Admin User', 'Created: 8/30/2020, 2:15 PM', 'Modified: 8/30/2020, 2:15 PM', and 'Type: Data Server'. Below this is a table with columns 'Name' and 'Modified', showing 'iextrading' with a modified date of '8/31/2020 9:53 AM'. The main area is titled 'Edit Progress DataDirect Autonomous REST connection'. It has tabs for 'General', 'Settings', 'Schemas', and 'Permissions'. Under 'Connection details', there are options for 'Authentication method' (Connect anonymously, Prompt for the user ID and password, Use an external namespace, Use the following signon) and a 'Test' button showing 'Success'. A 'Save' button is at the bottom right.

```
{  
  "Symbols.Symbols": "https://api.iextrading.com/1.0/ref-data/symbols",  
  "Stats.Stats_Records": "https://api.iextrading.com/1.0/stats/records",  
  "Stats.Stats_Historical": "https://api.iextrading.com/1.0/stats/historical/?date=201912",  
}
```

The screenshot shows the 'Data module' interface. On the left, there's a 'Sources' tree with 'SYMBOLS' expanded, showing 'Symbols', 'Data', and 'Stats'. The main area is a table with columns: 'Symbol', 'Name', 'Date', 'ESCALE', 'Type', and 'Sector'. The table contains various symbols like 'A', 'AABB', 'AACOM', etc., with their respective names and dates. A green arrow points from the code block above to the 'Stats' folder in the 'Sources' tree.

| Symbol | Name | Date | ESCALE | Type | Sector |
|--------|------------------------------|------------|--------|----------|--------|
| A | ABBREVIATED TECHNOLOGIES INC | 2020-10-05 | | unmapped | 2 |
| AAB | AACOM CORP | 2020-10-05 | | unmapped | 12 |
| AAC | AACORP INC | 2020-10-05 | | unmapped | 17 |
| AACD | ACQUISITION CORP | 2020-10-05 | | unmapped | 14 |
| AACF | ACQUISITION CORP | 2020-10-05 | | unmapped | 13 |
| AACG | ACQUISITION CORP | 2020-10-05 | | unmapped | 16 |
| AACI | ACQUISITION CORP | 2020-10-05 | | unmapped | 13 |
| AACJ | ACQUISITION CORP | 2020-10-05 | | unmapped | 16 |
| AACK | ACQUISITION CORP | 2020-10-05 | | unmapped | 13 |
| AACL | ACQUISITION CORP | 2020-10-05 | | unmapped | 16 |
| AACM | ACQUISITION CORP | 2020-10-05 | | unmapped | 13 |
| AACN | ACQUISITION CORP | 2020-10-05 | | unmapped | 16 |
| AACO | ACQUISITION CORP | 2020-10-05 | | unmapped | 13 |
| AACP | ACQUISITION CORP | 2020-10-05 | | unmapped | 16 |
| AACQ | ACQUISITION CORP | 2020-10-05 | | unmapped | 13 |
| AACR | ACQUISITION CORP | 2020-10-05 | | unmapped | 16 |
| AACS | ACQUISITION CORP | 2020-10-05 | | unmapped | 13 |
| AACU | ACQUISITION CORP | 2020-10-05 | | unmapped | 16 |
| AACV | ACQUISITION CORP | 2020-10-05 | | unmapped | 13 |
| AACW | ACQUISITION CORP | 2020-10-05 | | unmapped | 16 |
| AACX | ACQUISITION CORP | 2020-10-05 | | unmapped | 13 |
| AACY | ACQUISITION CORP | 2020-10-05 | | unmapped | 16 |
| AACZ | ACQUISITION CORP | 2020-10-05 | | unmapped | 13 |

Intent-based modeling

- When adding tables to a data module, you can engage this AI-based functionality to suggest the most appropriate tables for your module
- The selection of tables is based on keywords that you select

Discover related tables

Select keywords that describe your modeling objectives.

Q order method revenue product

Keywords limit: 75

Sources: gosales

Next Cancel

The image shows a word cloud interface for selecting keywords. The search query is 'order method revenue product'. The word cloud contains various terms such as 'product', 'quantity', 'sales', 'conversion', 'month', 'days', 'price', 'return', 'country', 'quarter', 'expected', 'shipped', 'target', 'city', 'code', 'brand', 'inventory', 'size', 'week', 'state', 'average', 'type', 'description', 'opening', 'zone', 'advised', 'inventory', 'ar', 'state', 'size', 'week', 'state', 'current', 'unit', 'sv', 'country', 'fi', 'quarter', 'expected', 'de', 'opening', 'average', 'type', 'description'. On the right, there are controls for 'Keywords limit' (set to 75) and 'Sources' (selected as 'gosales'). At the bottom, there are 'Next' and 'Cancel' buttons.

Discover related tables

Select one or more proposals that the system recommends. When you select multiple proposals, they are merged into one proposal.

Selected keywords: ordermethod revenue

Proposed data module

Q Search

- gosales 59%
 - Order Details
 - Order Header
 - Product
 - gosales [1] 59%
 - gosales [2] 59%

Advanced

Maximum number of proposals per source.

3

The image shows a 'Proposed data module' interface. The search query is 'ordermethod revenue'. The 'Proposed data module' is checked. The search results show 'gosales' with a 59% score and several sub-items: 'Order Details', 'Order Header', 'Product', 'gosales [1]' (59%), and 'gosales [2]' (59%). To the right, a diagram shows relationships between tables: 'Order Header' is connected to 'Order Details', 'Product', and 'Returned Item'. 'Order Details' is connected to 'Product' and 'Returned Item'. 'Product' is connected to 'Returned Item'. 'Returned Item' is connected to 'Return Reason'. 'Branch' is connected to 'Product Forecast'. 'Product Forecast' is connected to 'Branch'. At the bottom, there is an 'Advanced' section with a dropdown for 'Maximum number of proposals per source' set to 3.

Discover related tables

Select one or more proposals that the system recommends. When you select multiple proposals, they are merged into one proposal.

Selected keywords: order method product revenue line quantity

Proposed data module

Q Search

- gosales 100%
 - Branch
 - Order Details
 - Order Header
 - Product Type
 - Order Method
 - Product Forecast
 - Product Line
 - Country
 - Sales Region

The image shows a 'Proposed data module' interface. The search query is 'order method product revenue line quantity'. The 'Proposed data module' is checked. The search results show 'gosales' with a 100% score and several sub-items: 'Branch', 'Order Details', 'Order Header', 'Product Type', 'Order Method', 'Product Forecast', 'Product Line', 'Country', and 'Sales Region'. To the right, a diagram shows relationships between tables: 'Sales Region' is connected to 'Country'. 'Country' is connected to 'Branch'. 'Branch' is connected to 'Order Method'. 'Order Method' is connected to 'Order Header'. 'Order Header' is connected to 'Product Line'. 'Product Line' is connected to 'Product Type'. 'Product Type' is connected to 'Order Details'. 'Order Details' is connected to 'Product Type'. 'Sales Target' is connected to 'Product Type'. 'Product Type' is connected to 'Product Brand'. 'Product Brand' is connected to 'Product'. 'Product' is connected to 'Product Size Lookup'. 'Product Size Lookup' is connected to 'Product'. 'Product' is connected to 'Product Color Lookup'. At the bottom, there is an 'Advanced' section with a dropdown for 'Maximum number of proposals per source' set to 3.

Members in the tree (OLAP & relational)

- Available in dashboards, explore & reports
- Authors can expand fields in metadata tree to see individual values or members
- Data Modelers can control:
 - If dimensions can be expanded
 - If member search is available
 - Limits of how many members are shown

The screenshot shows a BI tool interface. On the left is a metadata tree for a 'Loyalty Module' with a search bar. The tree is expanded to show 'LoyaltyStatus' with members: Bronze, Elite, Gold, Platinum, Silver, and VIP. Below that is 'Product Line' with members: Home Theatre, Kitchen Appliances, Photography, Smart Electronics, and Video Games and Consoles. On the right is a report titled 'Loyalty Module' with a table of revenue by product line and loyalty status. Below the table is a 'Properties' panel with tabs for 'General', 'Relationships', and 'Navigation paths'. The 'General' tab is active, showing a 'Label' field set to 'Loyalty Module' and a 'Members display limit' dropdown set to 15. A 'Load More' button is visible below the dropdown.

| Revenue | Bronze | Elite | Gold | Summary |
|--------------------------|------------------|------------------|------------------|-------------------|
| Photography | 2,063,220 | 2,938,557 | 3,790,122 | 8,791,899 |
| Smart Electronics | 4,659 | 720,687 | 1,169,319 | 1,894,665 |
| Video Games and Consoles | 23,864 | 5,134,203 | 2,194,846 | 7,352,913 |
| Summary | 2,091,743 | 8,793,447 | 7,154,287 | 18,039,477 |

Properties

General Relationships Navigation paths

Label Loyalty Module

Members display limit

Set the limit

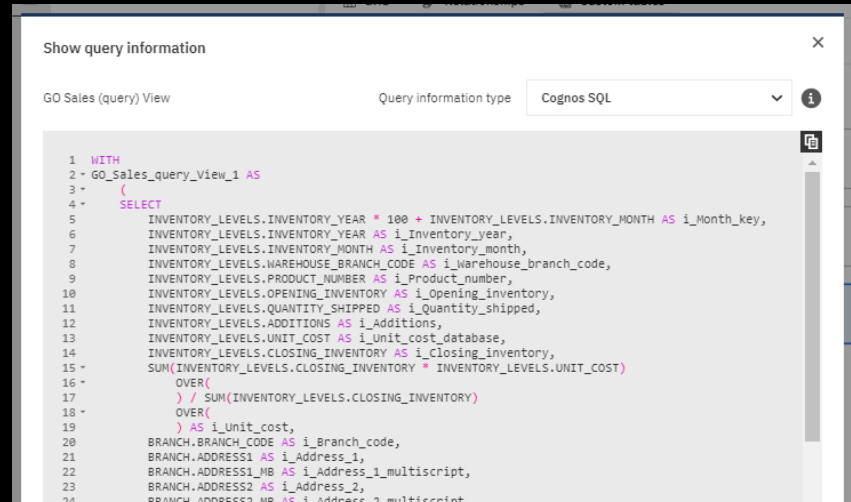
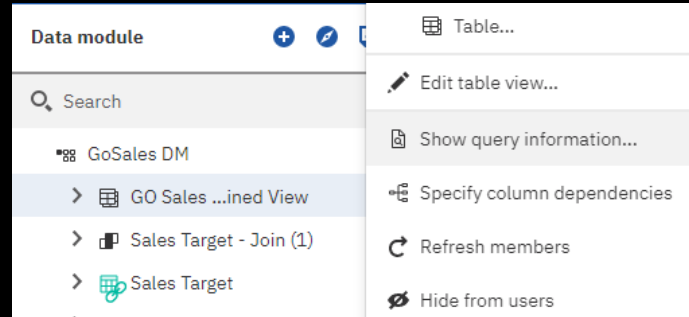
Members limit 15

More members Load More

Comments Load More

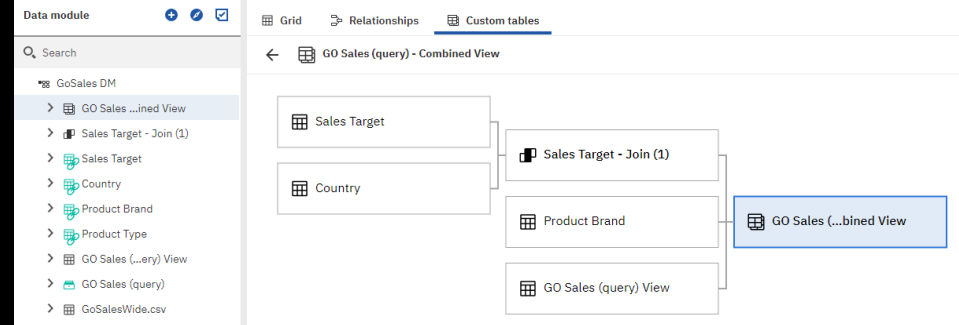
Query information (aka show the SQL)

- Allows a modeler to see the SQL behind a table/view without having to leave the modeling UI
- Helpful to debug - joins, unions, views, etc.



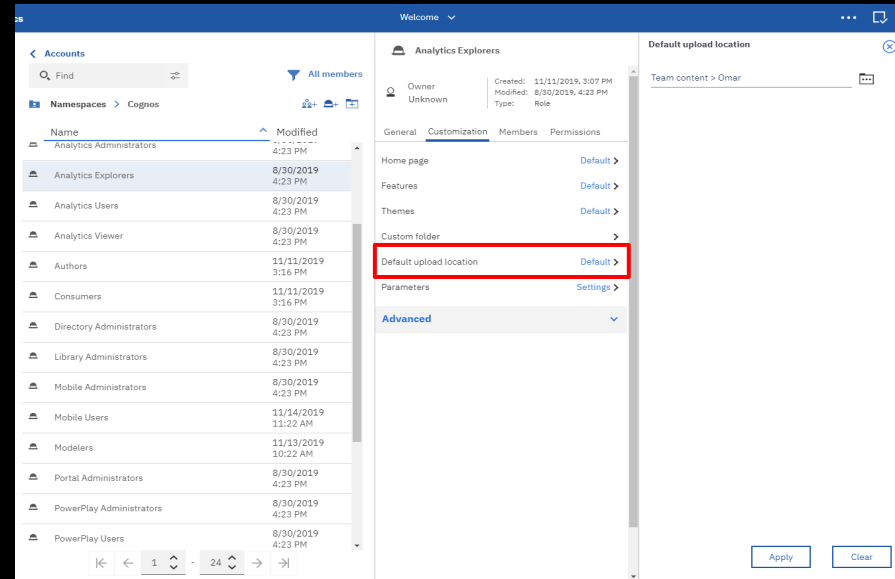
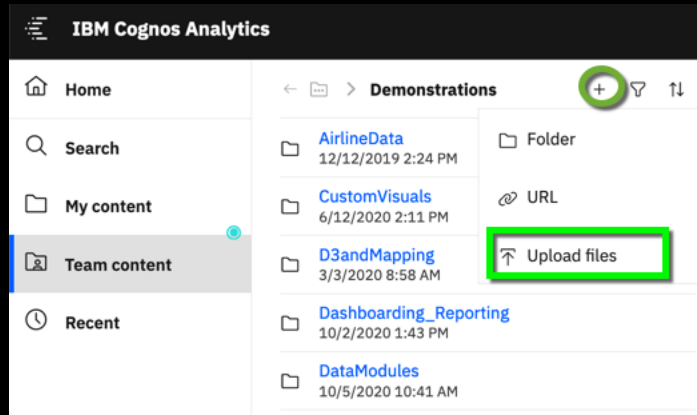
Enhanced UI for viewing custom tables

- Functionality to join tables existed in previous releases
- The added custom tables view gives a visual breakdown of how the different views and tables were created and which components they are composed of
- Helpful for impact analysis and debugging



Specify the default file upload location

- Control for default upload location is now available in role customizations
- Allows for higher shareability of files
- Upload files from within any folder location



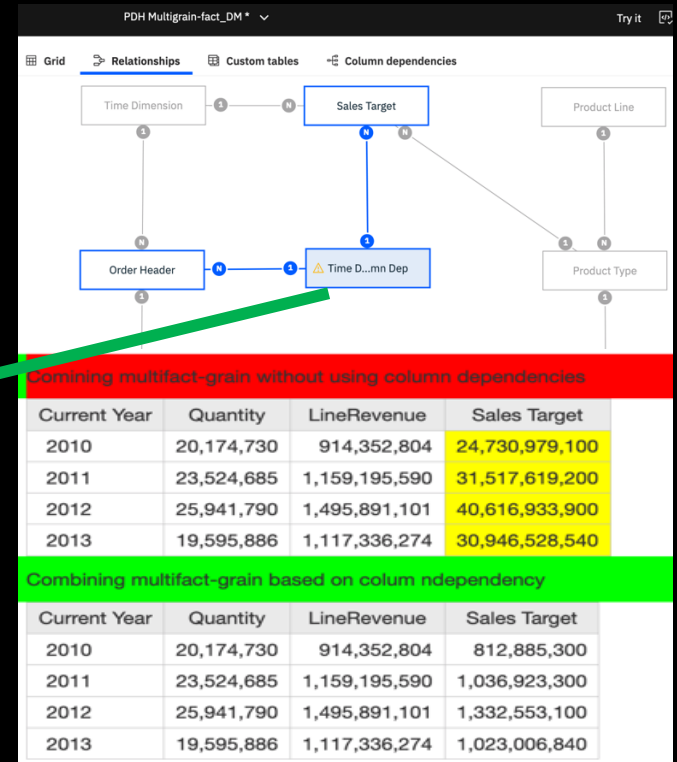
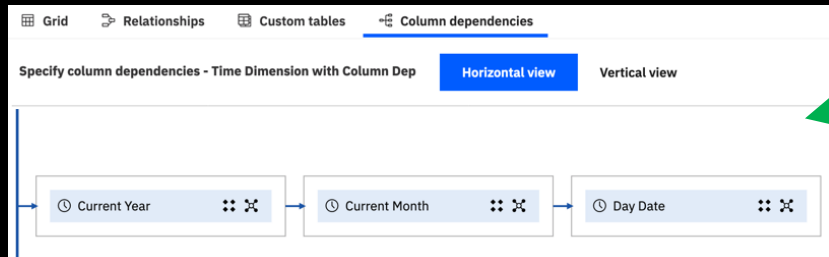
How to model multigrain/multifact

Typical use cases are when you have multiple facts on different grains:

- Sales data on product level
- Sales Target Data on product type level

Challenge: avoid double counting

- **Use column dependencies**

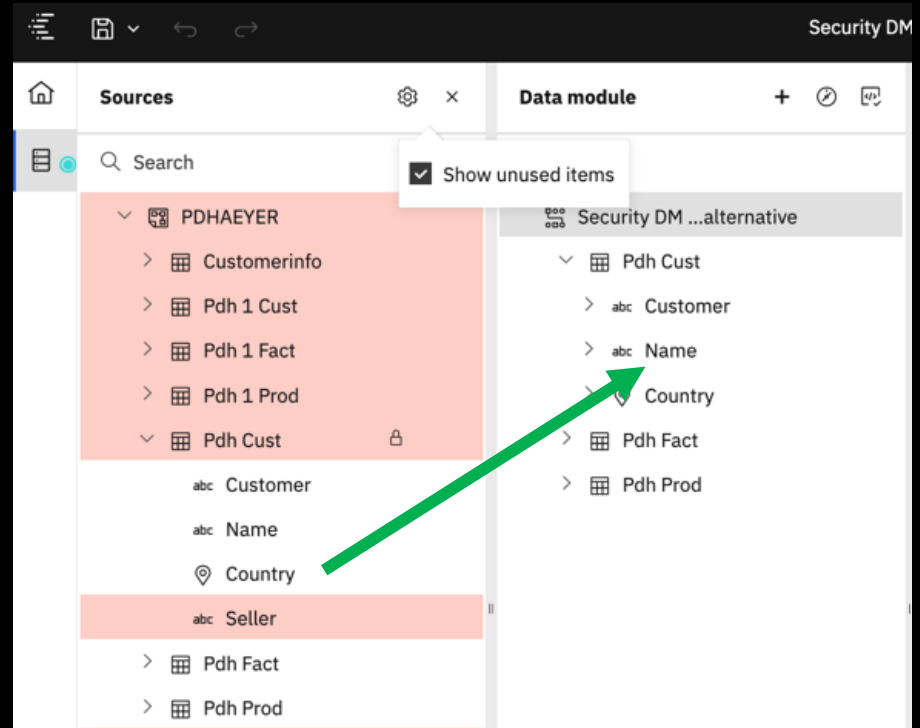
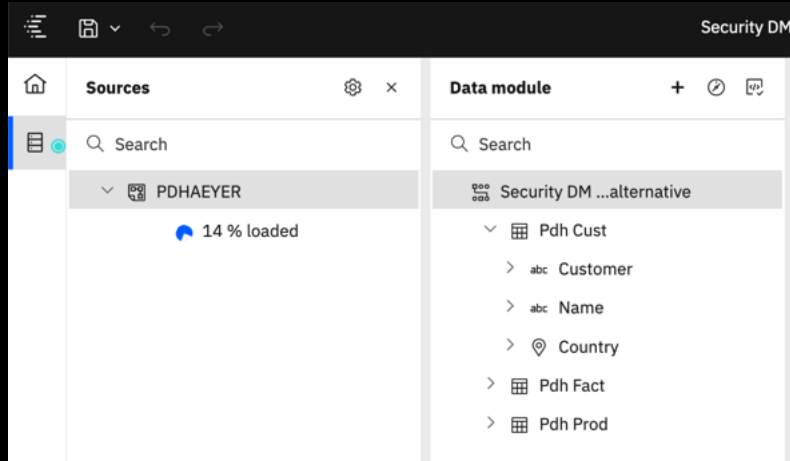


A nice video recording is available here:

<https://www.youtube.com/watch?v=MjzQBEsuqrM&list=PLzDSytIeSkT16fNpEVORNmO7isGMyjKfV&index=7&t=0s>

Easily add or update columns due to database changes in an existing data module

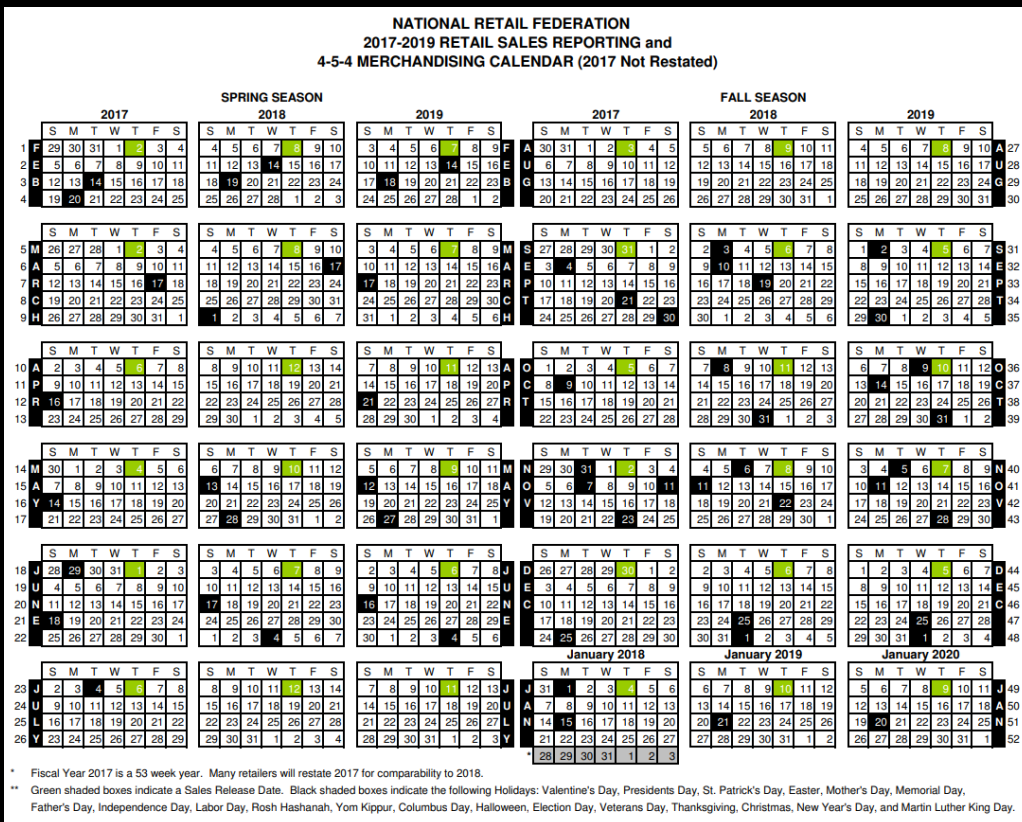
- Add individual columns directly from the source view to a data module
- Reload the schema metadata directly from a data module
- Easily differentiate unused items



4-5-4 Retail Calendar (in samples + SQL)

When creating data modules for relative date analysis, you can now use the sample 4-5-4 retail calendar.

The sample calendar is based on the National Retail Federation (NRF) [4-5-4 Calendar](#).



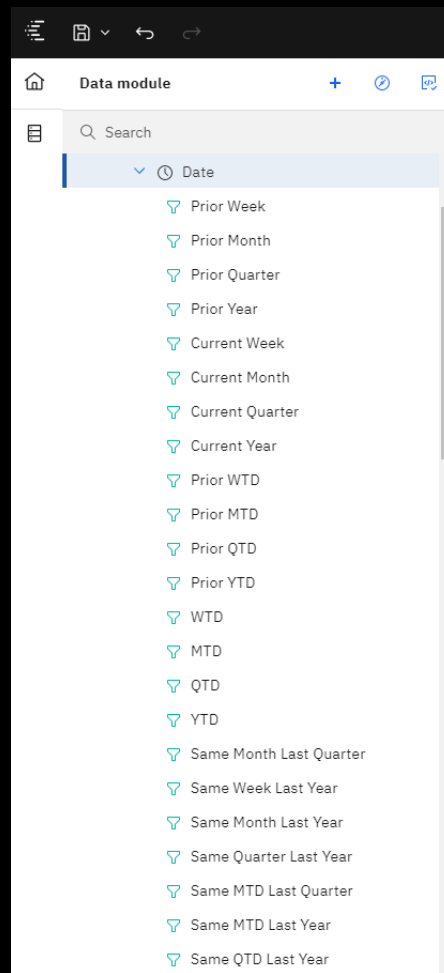
- Revenue
- Prior Week [Revenue]
- Prior Quarter [Revenue]
- Prior Year [Revenue]
- Current Week [Revenue]
- Current Month [Revenue]
- Current Quarter [Revenue]
- Current Year [Revenue]
- Prior WTD [Revenue]
- Prior QTD [Revenue]
- Prior YTD [Revenue]
- WTD [Revenue]
- MTD [Revenue]
- QTD [Revenue]
- YTD [Revenue]
- Same Month Last Quarter [Revenue]
- Same Week Last Year [Revenue]
- Same Month Last Year [Revenue]
- Same Quarter Last Year [Revenue]
- Same MTD Last Quarter [Revenue]
- Same MTD Last Year [Revenue]
- Same QTD Last Year [Revenue]

Additional out of the box Filters in all Calendars

Available in Gregorian, Fiscal, and 4-5-4 retail calendars – making it 23 filters in total

5 new filters:

- Current Week
- Prior week
- WTD
- Prior WTD
- Same week last Year



Some tips & tricks

to overcome

“hidden” limitations



Can you do multilingual modeling today?

Yes, except for metadata translation

Edit calculation

Name

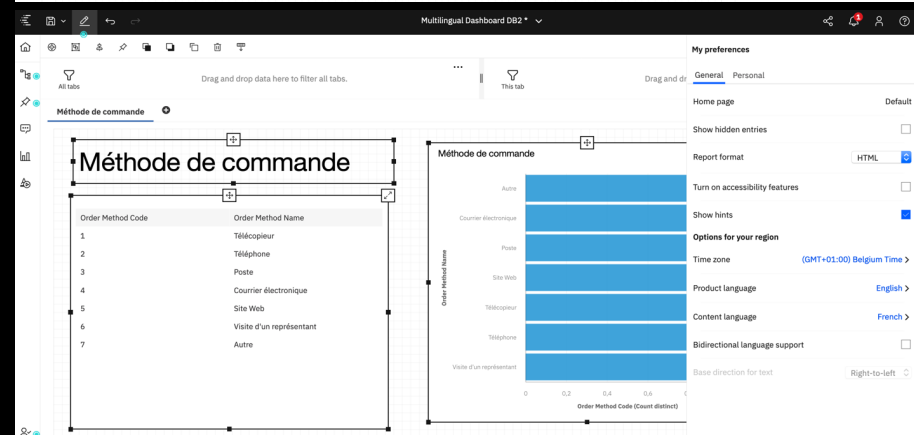
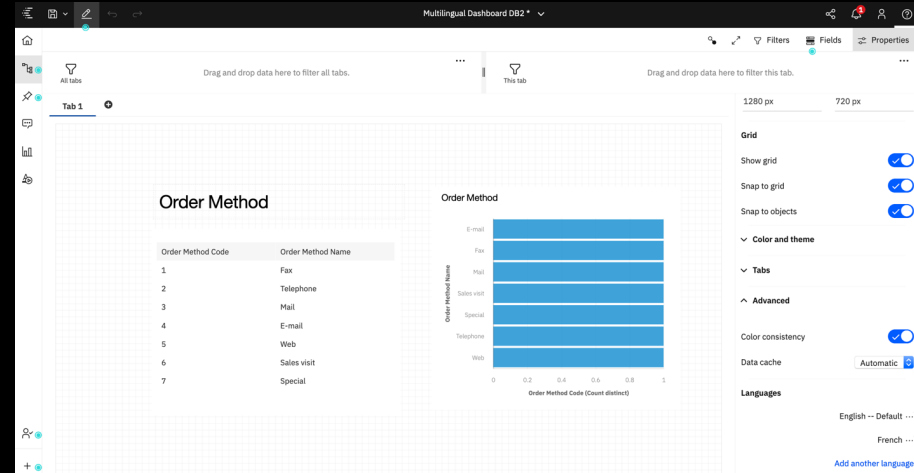
Components

Expression

```
1 //substring(upper(#sq($runLocale)#),1,2)
2 case substring(upper(#sq($runLocale)#),1,2)
3 when 'EN' then ORDER_METHOD.ORDER_METHOD_EN
4 when 'FR' then ORDER_METHOD.ORDER_METHOD_FR
5 ELSE ORDER_METHOD.ORDER_METHOD_NL
6 END
7
```

Preview (Execution time: 0.356 seconds)

| Order Method Name | Order Method En | Order Method Fr | Order Method Nl |
|-------------------|-----------------|--------------------------|-------------------|
| Mail | Mail | Poste | Post |
| Web | Web | Site Web | Internet |
| Sales visit | Sales visit | Visite d'un représentant | Vertegenwoordiger |
| Telephone | Telephone | Téléphone | Telefoon |
| Special | Special | Autre | Speciaal |



Import relationships of db's

With smarts, we all see automatic joining of sheets when relationship is detected

When uploading individual tables to a database, make sure to follow this procedure in order get automatic detection of tables:

- Make sure each table has a primary key
- For Fact tables, make sure to add foreign keys
- When loading metadata in CA, make sure options for foreign keys are selected

IBM Cognos Analytics Welcome ▾

Home

Search

My content

Team content

Recent

IBM DB2 warehouse pdh

Owner: administrator
Created: 3/20/2019, 6:00 AM
Modified: 9/20/2019, 12:14 AM
Type: Data Server

General **Connections** Permissions

| Name | Modified |
|-----------------------|-------------------|
| IBM DB2 warehouse pdh | 9/20/2019 2:52 AM |

IBM DB2 warehouse pdh

Owner: administrator
Created: 3/20/2019, 6:00 AM
Modified: 9/20/2019, 2:52 AM
Type: Connection

General Settings **Schemas** Permissions

| Status | Schema name | Load information |
|----------------------------------|-------------|----------------------|
| <input type="radio"/> | AUDIT | |
| <input checked="" type="radio"/> | DASH019437 | 6 / 37 tables loaded |
| <input type="radio"/> | DB2INST1 | |

DASH019437

Load options Tables

Cognos Analytics preloads information from your data servers to accelerate data preparation for reports, dashboards, and stories. Less information reduces the loading time, yet automated data preparation is also reduced.

Retrieve primary and foreign keys

Retrieve sample data

1000 rows (max 10,000)

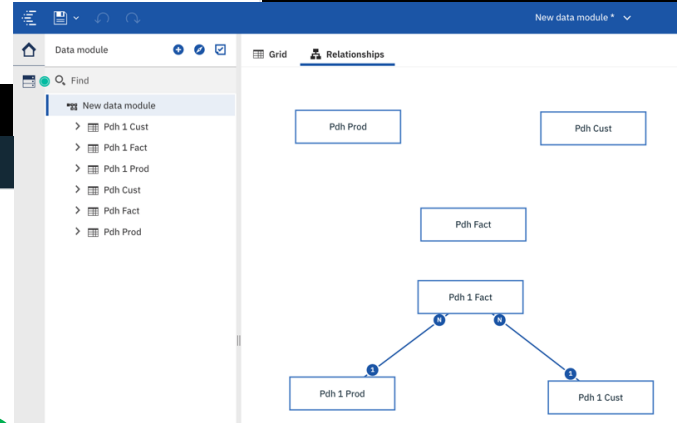
IBM Db2 Warehouse on Cloud Storage: 23%

RUN SQL

Run Script Edit Favorites New tab

```

1
2 --recognize db relationships between tables
3 drop table dash019437.PDH1_CUST;
4 create table dash019437.PDH1_CUST ( CUSTOMER VARCHAR(10) not null, NAME varchar(20) , COUNTRY varchar(20), PRIMARY KEY(CUSTOMER));
5 insert into dash019437.PDH1_CUST select * from dash019437.PDH_CUST;
6
7 drop table dash019437.PDH1_PROD;
8 create table dash019437.PDH1_PROD ( PRODUCT VARCHAR(10) not null, NAME varchar(20) , PRODUCT_TYPE varchar(20), PRIMARY KEY(PRODUCT));
9 insert into dash019437.PDH1_PROD select * from dash019437.PDH_PROD;
10
11 create table dash019437.PDH1_FACT ( CUSTOMER VARCHAR(10) not null, PRODUCT VARCHAR(10) not null, REVENUE integer , PLAN integer);
12 insert into dash019437.PDH1_FACT select * from dash019437.PDH_FACT;
13
14
15 ALTER TABLE dash019437.PDH1_FACT ADD FOREIGN KEY ((CUSTOMER) REFERENCES dash019437.PDH1_CUST (CUSTOMER) NOT ENFORCED;
16 ALTER TABLE dash019437.PDH1_FACT ADD FOREIGN KEY ((PRODUCT) REFERENCES dash019437.PDH1_PROD (PRODUCT) NOT ENFORCED;
17 --End of recognize db relationships between tables
  
```



How do automatic joins work?

The Auto-join algorithm adopts a diagnostic scoring approach to decide join columns between two tables. It employs a set of rules that applies to column combinations between two tables. Each rule produces a score. The score could be negative. The total score of all rules decides if a column combination qualifies to be a join column. Those rules are:

1. The similarity of two column names must exceed a minimum threshold. For example SalesCountryCode and CountryCode are highly similar and being considered to be matching.
2. Both columns belong to the same semantic category. For example, Employee
3. Both columns have the same semantic attribute. For example, both are have Usage = Identifier.
4. None of the column is a common row identifier. For example, rowid could be in every table
5. The data in two numeric columns must overlap.
6. The cardinality between the two columns is NOT be Many to Many.

A join relationship is created if any column combinations between two tables satisfy a minimum qualification score. Collected statistics is used to ensure cardinality is properly set when building the relationship. Those joins created by Auto-join algorithm are saved as inferred relationships.

From my anecdotal experience, the most common situation where no join is created when most folks expect a join to be created is the Many-to-Many scenario which like I said above, Cognos does not currently support in general and a bridge table workaround is far from trivial to do automatically. We will improve all this in the future.

Tip & trick: how to define security on fact table based on attribute in dimension table?

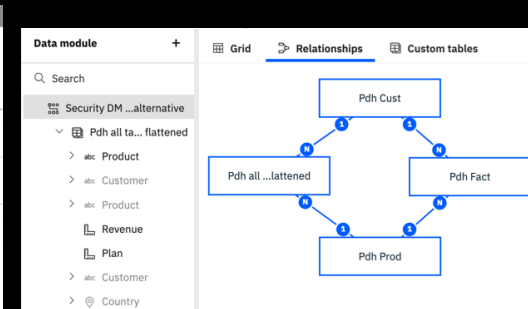
And avoid the message: In module "tempModule", the following query subjects are not joined: "Pdh Cust", "Pdh Fact".

The screenshot shows the SAP BW Security Manager interface for a data module named "Security DM...rehousePDH". The "Grid" tab is active, displaying a table with columns "Customer", "Name", and "Country". The table contains two rows: one for "Customer A" from "Belgium" and one for "Customer C" from "Belgium". The "Security filters" tab is also active, showing a filter named "Belgium Only" with the expression "((PDH_CUST.COUNTRY IN ('Belgium')))".

| Customer | Name | Country |
|----------|------------|---------|
| A | Customer A | Belgium |
| C | Customer C | Belgium |



The screenshot shows the "Set data security - Pdh Fact" dialog box in the SAP BW Security Manager. The "Filters" tab is active, showing a filter named "F_Country" with the expression "PDH_CUST.COUNTRY = 'Belgium'". The "Users, groups and roles" tab is also active, showing the user "Peter D'Haeyer (pdhaeyer)".



Solution : define a view on fact table in combination with all field required for security from other objects

Tip & trick: how to define security on fact table based on attribute in dimension table?

And avoid the message: In module "tempModule", the following query subjects are not joined: "Pdh Cust", "Pdh Fact".

What is the impact on existing dashboards, when security concepts are defined later on?

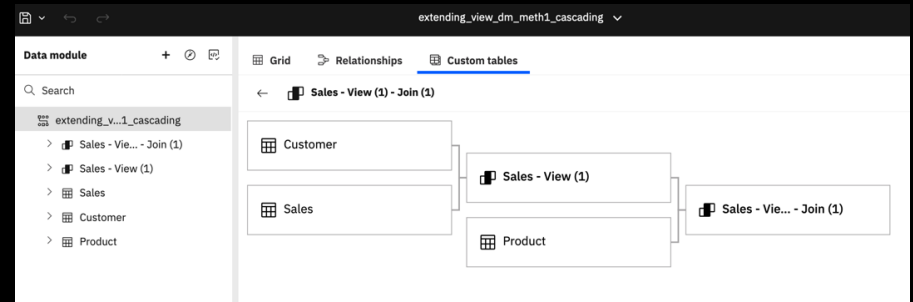
- Adding a new “fact secure view” will have implications on existing dashboards
 - Relink the (new) datamodule in your dashboard if name has changed
 - Reassign all field manually
- Or, If you keep the same DM name, make sure to rename your view to the original Fact table (including identifiers) and copy orig fact to new name and identifier => in this case, **no changes to dashboard needed**
- Or use CTRL + / + Q and do manual replacements (tricky)

The screenshot shows the Qlik Sense interface for a data module named "Security DM DB2WarehousePDH". The main view is a table with columns: Customer, Country, Revenue, Plan, and Product. The table contains four rows of data. A green box highlights the table. Below the table, a relationship diagram shows "Pdh Fact" connected to "Pdh Cust" and "Pdh Prod". A red box highlights a "Custom tables" window showing a list of tables including "Pdh Fact". The right sidebar shows the "Properties" panel for "Pdh Fact", with the "Identifier" set to "PDH_FACT".

| Customer | Country | Revenue | Plan | Product |
|----------|---------|---------|------|---------|
| C | Belgium | 2300 | 3000 | P3 |
| A | Belgium | 2000 | 3000 | P1 |
| A | Belgium | 2400 | 3000 | P1 |

How extending an existing view with extra tables ? ... with no / minimal impact on existing dashboards & reports

- Add new table in a cascading view



- Use calc fields to reference new table fields

The screenshot shows a data model interface with a left-hand navigation pane and a main workspace. The navigation pane is titled 'Data module' and contains a search bar and a tree view for 'extending_v...meth2_calcs'. The tree view includes 'Sales - View (1)' with a sub-tree containing '# Row Id', 'Date', 'Product', 'Quantity', 'Unit Price', 'Cost', 'Customer', 'Name', 'Country', 'Product Name', and 'Product Category'. The main workspace is titled 'Sales - View (1)' and displays a diagram where 'Sales' and 'Customer' are connected to 'Sales - View (1)'. A green arrow points from the 'Product Name' field in the left pane to the 'Expression' field in the 'Edit calculation' dialog. The dialog shows 'Name' as 'Product Name' and 'Expression' as '1 Product.productname'. A 'New' menu is open, showing options like 'Calculation...', 'Relationship...', 'Filter...', 'Table...', 'Edit joined table...', 'Manage filters', and 'Properties'.

How extending an existing view with extra tables ? ... with no / minimal impact on existing dashboards & reports

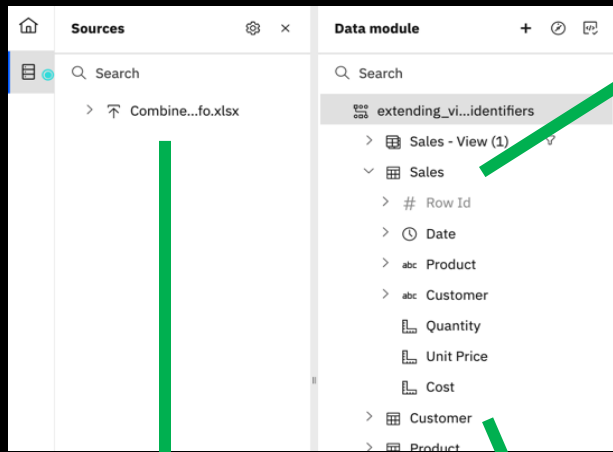
- Create new view and change identifiers... and datamodule xml (CTRL + / + Q)

The screenshot shows the SAP Data Builder interface for a data module named 'extending_view_dm_meth3_identifiers'. The 'Custom tables' section is active, showing a hierarchy of tables: 'Sales - View (2)' is the parent, and 'Product', 'Customer', and 'Sales' are its children. The 'Properties' pane for 'Sales - View (2)' is open, showing the 'General' tab. The 'Identifier' property is set to 'Sales_View_2' (circled in green). The 'Advanced' tab is also visible, showing 'Usage' set to 'Select an option' and 'Item list' set to 'Select an option'. In the left-hand 'Data module' tree, the 'Country' field is circled in red.

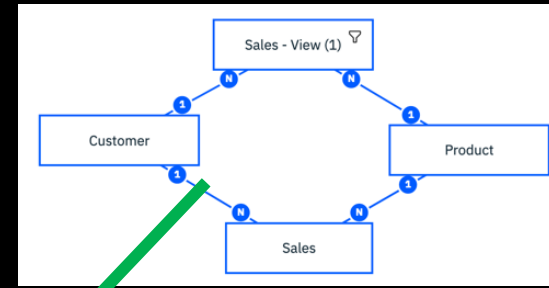
The screenshot shows the XML configuration for a view in the SAP Data Builder. The 'Sales - View (1)' is selected in the left-hand tree (circled in green). The XML configuration is displayed on the right, with a green circle around the 'ref' field and an orange box around the 'filter' field. The 'ref' field is set to 'Sales' and 'Customer'. The 'filter' field is set to a complex expression: `"expression": "(Sales_View_1.Country IN ('Belgium'))", "identifier": "Sales_View_1_Country", "label": "Country", "property": { "name": "_MUI_FilterSpec", "value": "{\n \"attributeUniqueNames\": [\"Sales_View_1_Country\"], \"filterType\": null, \"dataType\": \"string\", \"rangeSubtype\": null, \"condition\": null, \"invert\": false, \"fromValue\": null, \"toValue\": null, \"selectedValues\": [\"Belgium\"], \"excludedValues\": [], \"version\": \"1.0\"}"}`. The 'Country' field in the left-hand tree is also circled in red.

Tip & trick: data module xml structure

If you can't hit CTRL+/,+Q, change your keyboard settings to british



```
"querySubject": [
  {
    "ref": [
      "M1.Sales"
    ],
    "item": [
      {
        "queryItem": {
          "expression": "_row_id",
          "usage": "identifier",
          "datatype": "BIGINT",
          ...
        }
      },
      {
        "queryItem": {
          "expression": "Date_",
          ...
          "idForExpression": "Sales.Date_"
        }
      }
    ],
    "identifier": "Sales",
    "label": "Sales",
    "property": [
      {
        "name": "_MUJ_diagramNodePosition",
        "value": "{\\x\\":591.5403214749076,\\y\\":478.7490003330539}"
      }
    ],
    "propertyOverride": [
      "NEW"
    ],
    "idForExpression": "Sales"
  },
  {
    "ref": [
      "M1.Customer"
    ],
    "item": [
      {
        "queryItem": {
          "expression": "Date_",
          ...
          "idForExpression": "Sales.Date_"
        }
      }
    ],
    "identifier": "Customer",
    "label": "Customer",
    "property": [
      {
        "name": "_MUJ_diagramNodePosition",
        "value": "{\\x\\":591.5403214749076,\\y\\":478.7490003330539}"
      }
    ],
    "propertyOverride": [
      "NEW"
    ],
    "idForExpression": "Customer"
  }
],
"relationship": [
  {
    "left": {
      "ref": "Customer",
      "minCard": "one",
      "maxCard": "one"
    },
    "right": {
      "ref": "Sales",
      "minCard": "one",
      "maxCard": "many"
    },
    "link": [
      {
        "leftRef": "Customer",
        "rightRef": "Customer",
        "comparisonOperator": "equalTo"
      }
    ],
    "joinFilterType": "none",
    "identifier": "Sales_Customer",
    "description": "Sales<-->Customer",
    "label": "Customer<-->Sales",
    "property": [
      {
        "name": "SystemDiscovered",
        "value": "true"
      }
    ]
  },
  {
    "left": {
      "ref": "Sales",
      "minCard": "one",
      "maxCard": "one"
    },
    "right": {
      "ref": "Product",
      "minCard": "one",
      "maxCard": "one"
    },
    "link": [
      {
        "leftRef": "Sales",
        "rightRef": "Product",
        "comparisonOperator": "equalTo"
      }
    ],
    "joinFilterType": "none",
    "identifier": "Product_Sales",
    "description": "Product<-->Sales",
    "label": "Sales<-->Product",
    "property": [
      {
        "name": "SystemDiscovered",
        "value": "true"
      }
    ]
  }
],
"metadataTreeView": [
  {
    "folderItem": [
      {
        "ref": "Sales_View_1"
      },
      {
        "ref": "Sales"
      },
      {
        "ref": "Customer"
      },
      {
        "ref": "Product"
      }
    ]
  }
],
"expressionLocale": "en-us",
```



```
"relationship": [
  {
    "left": {
      "ref": "Customer",
      "minCard": "one",
      "maxCard": "one"
    },
    "right": {
      "ref": "Sales",
      "minCard": "one",
      "maxCard": "many"
    },
    "link": [
      {
        "leftRef": "Customer",
        "rightRef": "Customer",
        "comparisonOperator": "equalTo"
      }
    ],
    "joinFilterType": "none",
    "identifier": "Sales_Customer",
    "description": "Sales<-->Customer",
    "label": "Customer<-->Sales",
    "property": [
      {
        "name": "SystemDiscovered",
        "value": "true"
      }
    ]
  },
  {
    "left": {
      "ref": "Sales",
      "minCard": "one",
      "maxCard": "one"
    },
    "right": {
      "ref": "Product",
      "minCard": "one",
      "maxCard": "one"
    },
    "link": [
      {
        "leftRef": "Sales",
        "rightRef": "Product",
        "comparisonOperator": "equalTo"
      }
    ],
    "joinFilterType": "none",
    "identifier": "Product_Sales",
    "description": "Product<-->Sales",
    "label": "Sales<-->Product",
    "property": [
      {
        "name": "SystemDiscovered",
        "value": "true"
      }
    ]
  }
],
"metadataTreeView": [
  {
    "folderItem": [
      {
        "ref": "Sales_View_1"
      },
      {
        "ref": "Sales"
      },
      {
        "ref": "Customer"
      },
      {
        "ref": "Product"
      }
    ]
  }
],
"expressionLocale": "en-us",
```

```
"metadataTreeView": [
  {
    "folderItem": [
      {
        "ref": "Sales_View_1"
      },
      {
        "ref": "Sales"
      },
      {
        "ref": "Customer"
      },
      {
        "ref": "Product"
      }
    ]
  }
],
"expressionLocale": "en-us",
```





Setting up custom Relative times (! changes as of 11.1.7 !)

- Standard documentation available here
https://www.ibm.com/support/knowledgecenter/en/SSEP7J_11.1.0/com.ibm.swg.ba.cognos.ca_mdlg.doc/c_ca_set_rel_date_analysis.html

Step by step :

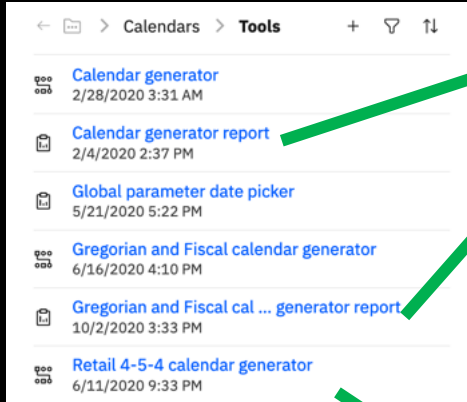
- Copy paste all required assets from “Team Folder/tools” to your location. Modify as required:
- Add custom fields to report output (generated csv) and/or directly in table sql of corresp. Module
- Relink generated csv into your New Custom Calendar Module (evt copied from existing)
- Add Custom Relative filters to New Custom Calendar Module
- Hide all entries and save your New Custom Calendar Module
- Proceed as usual to add rel times to your data source



| | | |
|--|---|-------------------|
|  | 1 PDH Gregorian and Fiscal calendar generator | 9/29/2020 3:11 PM |
|  | 2 PDH Gregorian and Fiscal ca ... generator report | 10/2/2020 3:36 PM |
|  | 3 PDH Gregorian and Fisc ... generator report.csv | 9/29/2020 3:01 PM |
|  | 4 PDH calendar new | 9/29/2020 3:12 PM |
|  | 5 PDH Sales DM new | 9/29/2020 3:14 PM |

Modifying the Calendar generator report (as of 11.1.7)

New calendar 4-5-4 (retail) available, requiring new ways of building your own Custom Calendars



Calendar generator report : old method with prompted report to create csv for x years as of you chosen starting date

Gregorian and Fiscal calendar generator report:

- new method to build a fixed period calendar (no prompt any more)
- Date ranges are now defined in the sql table of corresponding datamodule
- You have to modify the sql table in DM (save all as your own objects)
- You can still insert prompt code from orig method into sql table of DM
- Does now contain “week” relative time concepts

Retail 4-5-4 calendar generator:

- No report available to generate csv
- More changes needed in sql table code in data module
- Please follow all instructions as described in std documentation
- https://www.ibm.com/support/knowledgecenter/en/SSEP7J_11.1.0/com.ibm.swg.ba.cognos.ca_mdlg.doc/t_ca_mdlg_cust_retail_cal.html

Adding custom fields the Calendar generator report

Example Prior 6 months date (P6M_TheDate)

IBM Cognos Analytics navigation pane showing the 'Calendar generator report PDH' selected. The pane includes a search bar, 'My content', 'Team content', and 'Recent' sections. The 'Calendar generator report PDH' is listed with a timestamp of 9/20/2019 8:50 AM.

Calendar generator report PDH

To generate a CSV file based on this data, click the Run icon in the Application Bar and select Run CSV.
Note. Do not modify any of the column names in this report.

| TheDate | PD_TheDate | ND_TheDate | dYear | PY_TheDate | NY_TheDate | dQuarter | PQ_TheDate | NQ_TheDate | dMonth | PM_TheDate | NM_TheDate | P6M_TheDate |
|--------------|--------------|--------------|-------------|--------------|--------------|-------------|--------------|--------------|-------------|--------------|--------------|--------------|
| Jan 1, 2015 | Dec 31, 2014 | Jan 2, 2015 | Jan 1, 2015 | Jan 1, 2014 | Jan 1, 2016 | Jan 1, 2015 | Oct 1, 2014 | Apr 1, 2015 | Jan 1, 2015 | Dec 1, 2014 | Feb 1, 2015 | Jul 1, 2014 |
| Jan 2, 2015 | Jan 1, 2015 | Jan 3, 2015 | Jan 1, 2015 | Jan 2, 2014 | Jan 2, 2016 | Jan 1, 2015 | Oct 2, 2014 | Apr 2, 2015 | Jan 1, 2015 | Dec 2, 2014 | Feb 2, 2015 | Jul 2, 2014 |
| Jan 3, 2015 | Jan 2, 2015 | Jan 4, 2015 | Jan 1, 2015 | Jan 3, 2014 | Jan 3, 2016 | Jan 1, 2015 | Oct 3, 2014 | Apr 3, 2015 | Jan 1, 2015 | Dec 3, 2014 | Feb 3, 2015 | Jul 3, 2014 |
| Jan 4, 2015 | Jan 3, 2015 | Jan 5, 2015 | Jan 1, 2015 | Jan 4, 2014 | Jan 4, 2016 | Jan 1, 2015 | Oct 4, 2014 | Apr 4, 2015 | Jan 1, 2015 | Dec 4, 2014 | Feb 4, 2015 | Jul 4, 2014 |
| Jan 5, 2015 | Jan 4, 2015 | Jan 6, 2015 | Jan 1, 2015 | Jan 5, 2014 | Jan 5, 2016 | Jan 1, 2015 | Oct 5, 2014 | Apr 5, 2015 | Jan 1, 2015 | Dec 5, 2014 | Feb 5, 2015 | Jul 5, 2014 |
| Jan 6, 2015 | Jan 5, 2015 | Jan 7, 2015 | Jan 1, 2015 | Jan 6, 2014 | Jan 6, 2016 | Jan 1, 2015 | Oct 6, 2014 | Apr 6, 2015 | Jan 1, 2015 | Dec 6, 2014 | Feb 6, 2015 | Jul 6, 2014 |
| Jan 7, 2015 | Jan 6, 2015 | Jan 8, 2015 | Jan 1, 2015 | Jan 7, 2014 | Jan 7, 2016 | Jan 1, 2015 | Oct 7, 2014 | Apr 7, 2015 | Jan 1, 2015 | Dec 7, 2014 | Feb 7, 2015 | Jul 7, 2014 |
| Jan 8, 2015 | Jan 7, 2015 | Jan 9, 2015 | Jan 1, 2015 | Jan 8, 2014 | Jan 8, 2016 | Jan 1, 2015 | Oct 8, 2014 | Apr 8, 2015 | Jan 1, 2015 | Dec 8, 2014 | Feb 8, 2015 | Jul 8, 2014 |
| Jan 9, 2015 | Jan 8, 2015 | Jan 10, 2015 | Jan 1, 2015 | Jan 9, 2014 | Jan 9, 2016 | Jan 1, 2015 | Oct 9, 2014 | Apr 9, 2015 | Jan 1, 2015 | Dec 9, 2014 | Feb 9, 2015 | Jul 9, 2014 |
| Jan 10, 2015 | Jan 9, 2015 | Jan 11, 2015 | Jan 1, 2015 | Jan 10, 2014 | Jan 10, 2016 | Jan 1, 2015 | Oct 10, 2014 | Apr 10, 2015 | Jan 1, 2015 | Dec 10, 2014 | Feb 10, 2015 | Jul 10, 2014 |
| Jan 11, 2015 | Jan 10, 2015 | Jan 12, 2015 | Jan 1, 2015 | Jan 11, 2014 | Jan 11, 2016 | Jan 1, 2015 | Oct 11, 2014 | Apr 11, 2015 | Jan 1, 2015 | Dec 11, 2014 | Feb 11, 2015 | Jul 11, 2014 |
| Jan 12, 2015 | Jan 11, 2015 | Jan 13, 2015 | Jan 1, 2015 | Jan 12, 2014 | Jan 12, 2016 | Jan 1, 2015 | Oct 12, 2014 | Apr 12, 2015 | Jan 1, 2015 | Dec 12, 2014 | Feb 12, 2015 | Jul 12, 2014 |
| Jan 13, 2015 | Jan 12, 2015 | Jan 14, 2015 | Jan 1, 2015 | Jan 13, 2014 | Jan 13, 2016 | Jan 1, 2015 | Oct 13, 2014 | Apr 13, 2015 | Jan 1, 2015 | Dec 13, 2014 | Feb 13, 2015 | Jul 13, 2014 |
| Jan 14, 2015 | Jan 13, 2015 | Jan 15, 2015 | Jan 1, 2015 | Jan 14, 2014 | Jan 14, 2016 | Jan 1, 2015 | Oct 14, 2014 | Apr 14, 2015 | Jan 1, 2015 | Dec 14, 2014 | Feb 14, 2015 | Jul 14, 2014 |
| Jan 15, 2015 | Jan 14, 2015 | Jan 16, 2015 | Jan 1, 2015 | Jan 15, 2014 | Jan 15, 2016 | Jan 1, 2015 | Oct 15, 2014 | Apr 15, 2015 | Jan 1, 2015 | Dec 15, 2014 | Feb 15, 2015 | Jul 15, 2014 |
| Jan 16, 2015 | Jan 15, 2015 | Jan 17, 2015 | Jan 1, 2015 | Jan 16, 2014 | Jan 16, 2016 | Jan 1, 2015 | Oct 16, 2014 | Apr 16, 2015 | Jan 1, 2015 | Dec 16, 2014 | Feb 16, 2015 | Jul 16, 2014 |
| Jan 17, 2015 | Jan 16, 2015 | Jan 18, 2015 | Jan 1, 2015 | Jan 17, 2014 | Jan 17, 2016 | Jan 1, 2015 | Oct 17, 2014 | Apr 17, 2015 | Jan 1, 2015 | Dec 17, 2014 | Feb 17, 2015 | Jul 17, 2014 |
| Jan 18, 2015 | Jan 17, 2015 | Jan 19, 2015 | Jan 1, 2015 | Jan 18, 2014 | Jan 18, 2016 | Jan 1, 2015 | Oct 18, 2014 | Apr 18, 2015 | Jan 1, 2015 | Dec 18, 2014 | Feb 18, 2015 | Jul 18, 2014 |
| Jan 19, 2015 | Jan 18, 2015 | Jan 20, 2015 | Jan 1, 2015 | Jan 19, 2014 | Jan 19, 2016 | Jan 1, 2015 | Oct 19, 2014 | Apr 19, 2015 | Jan 1, 2015 | Dec 19, 2014 | Feb 19, 2015 | Jul 19, 2014 |
| Jan 20, 2015 | Jan 19, 2015 | Jan 21, 2015 | Jan 1, 2015 | Jan 20, 2014 | Jan 20, 2016 | Jan 1, 2015 | Oct 20, 2014 | Apr 20, 2015 | Jan 1, 2015 | Dec 20, 2014 | Feb 20, 2015 | Jul 20, 2014 |

Data item expression - P6M_TheDate

Name: P6M_TheDate

Available Components:

- Calendar generator
- Calendar Generator

Expression Definition:

```
_add_months ([C].[C_Calendar_generator].[Calendar_Generator].[PM_TheDate],-5 )
```

Information:

OK Cancel

Modifying custom Calendar Data module

Example adding custom filter Rolling 6 Month (R6M_Extended)

Sources

- 3 PDH Greg...report.csv
- PDH Gregor...rt(1).csv
 - Row Id
 - TheDate
 - PD_TheDate
 - ND_TheDate
 - dYear
 - PY_TheDate
 - NY_TheDate
 - dQuarter
 - PQ_TheDate
 - NQ_TheDate
 - dMonth
 - PM_TheDate
 - NM_TheDate
 - dWeek
 - PW_TheDate
 - NW_TheDate
 - P6M_TheDate
 - Week-2

Data module

- 4 PDH calendar new
 - PDH Calendar
 - WTD
 - Prior WTD
 - Current Week
 - Prior Week
 - R6M Extended
 - R6M Classic
 - Prior Month
 - Prior Quarter
 - Prior Year
 - Current Month
 - Current Quarter
 - Current Year
 - Prior MTD
 - Prior QTD
 - Prior YTD
 - MTD
 - QTD
 - YTD

Edit filter (read-only)

Name: R6M extended field

```
1 // This is a template expression that is used by the column property 'Lookup reference'.
2 // To pass validation, the line below must remain as a comment. Do not remove the forward slashes.
3 // validate: 1 = 1
4 #$_this.parent.idForExpression# >=
5   queryValue($_this.parent.split.ref + '.dMonth',
6     $_this.parent.split.ref + '.TheDate' + ' +
7     queryValue($_this.parent.split.ref + '.P6M_TheDate',
8       $_this.parent.split.ref + '.TheDate' + ' + $_as_of_date)
9   )#
10 AND
11 #$_this.parent.idForExpression# <=
12   queryValue($_this.parent.split.ref + '.PM_TheDate',
13     $_this.parent.split.ref + '.TheDate' + ' + $_as_of_date)
14
```

Edit filter (read-only)

Name: R6M Classic

```
1 // This is a template expression that is used by the column property 'Lookup reference'.
2 // To pass validation, the line below must remain as a comment. Do not remove the forward slashes.
3 // validate: 1 = 1
4 #$_this.parent.idForExpression# >=
5   queryValue($_this.parent.split.ref + '.dMonth',
6     $_this.parent.split.ref + '.TheDate' + ' +
7     queryValue($_this.parent.split.ref + '.PM_TheDate',
8       $_this.parent.split.ref + '.TheDate' + ' +
9     queryValue($_this.parent.split.ref + '.PM_TheDate',
10      $_this.parent.split.ref + '.TheDate' + ' +
11     queryValue($_this.parent.split.ref + '.PM_TheDate',
12      $_this.parent.split.ref + '.TheDate' + ' +
13     queryValue($_this.parent.split.ref + '.PM_TheDate',
14      $_this.parent.split.ref + '.TheDate' + ' +
15     queryValue($_this.parent.split.ref + '.PM_TheDate',
16      $_this.parent.split.ref + '.TheDate' + ' +
17     queryValue($_this.parent.split.ref + '.PM_TheDate',
18      $_this.parent.split.ref + '.TheDate' + ' + $_as_of_date)
19   )
20 )
21 )
22 )
23 )
24 )#
25 AND
26 #$_this.parent.idForExpression# <
27   queryValue($_this.parent.split.ref + '.dMonth',
28     $_this.parent.split.ref + '.TheDate' + ' + $_as_of_date)
29
```

Data module

- PDH Great outdoors data module
 - PDH Calendar
 - Branches
 - Retailers
 - Order Methods
 - Products
 - Returns
 - Sales Representatives
 - Sales
 - Country Code
 - Order Number
 - Retailer Code
 - Product Number
 - Order Method Code
 - Sales Order Code
 - Date
 - Year
 - Quarter
 - Month
 - Quantity
 - Unit Cost
 - Unit Price
 - Unit Sale Price
 - Revenue
 - R6M Extended [Revenue]

Modifying the Calendar generator data module (as of 11.1.7)

The screenshot displays a data modeling tool interface. At the top, the title bar reads "1 PDH Gregorian and Fiscal calendar generator". The main workspace is divided into a left-hand "Data module" pane and a right-hand "Edit table" pane.

Data module pane: Shows a tree view under "1 PDH Gregor...r generator". A sub-folder "sql Calendar Generator" is expanded, listing several tables: TheDate, PD_TheDate, ND_TheDate, dYear, PY_TheDate, NY_TheDate, dQuarter, PQ_TheDate, NQ_TheDate, dMonth, PM_TheDate, NM_TheDate, dWeek, PW_TheDate, and NW_TheDate.

Edit table pane: Shows the configuration for the "Calendar Generator" table. The "Name" field is "Calendar Generator" and the "SQL Type" is "Cognos SQL". The "Source" field is a dropdown menu currently showing "Select an option". The "Expression" field contains the following SQL code:

```
9 /* change pdh --- added old prompt and replace lines 9-23*/
10 with
11- calendar_settings (fiscalStartDate, numOfYears) as (
12   select *
13   from ( values ( # prompt('pStartDateGeneration', 'date')#, # prompt('pNumberOfYears', 'integer')# ) ) T2 ( fiscalStartDate, numOfYears )
14 ),
15- gen_rows ( R ) as (
16   select c1 from ( values ( 0 ) ) t (c1)
17   union all
18   /* X * 365 days in a year */
19   select R + 1 from gen_rows where ( ((select numOfYears from calendar_settings) * 365) + (cast ((select numOfYears from calendar_settings)/
20 ),
21- dates ( D ) as (
22-   select _add_days ( cast (_make_timestamp( extract ( year from (select fiscalStartDate from calendar_settings) ),
23     extract ( month from (select fiscalStartDate from calendar_settings) ),
24     1) as date) , R ) from gen_rows
25 )
26 /* pdh end of */
27- select
```

Usage with _as_of_date

Works by default in Reports

make sure you add an explicit optional filter in your report in the query in order to have "as_of_date" available to change

Combined Sales Report

1/3 Alerts | What's New | To read about what's new in Reporting, click More Info.

As of Date : 2018-12-02

| Product | YTD [Quantity] | Prior YTD [Quantity] | R6M Extended [Quantity] |
|--------------------------|----------------|----------------------|-------------------------|
| P1 | 179 | 178 | 32 |
| P2 | 69 | 32 | 36 |
| P3 | 101 | 127 | 50 |
| P4 | 175 | 85 | 120 |
| P5 | 82 | 37 | 55 |
| P6 | 385 | 605 | 140 |
| Overall - Summary | 991 | 1,064 | 433 |

My parameters

_as_of_date: 2018-12-02

Report > Queries > Query1

Data Items

- Product
- YTD [Quantity]
- Prior YTD [Quantity]
- R6M Extended [Quantity]
- Summary(R6M Extended [Quantity])

Detail Filters

- [Date] <= ?_as_of_date?

Summary Filters

You can also leverage this variable within dashboards

Name: ASOFDATE

Components

- Sales
- ASOFDATE
- Yearmonth
- Row Id

Expression

```
1 #ParamValue ( '_as_of_date' )#
2
```

PDH Sales dashboard with Cust Ret Time As of *

1/2 Alerts | What's New | To read about what's new in Dashboards, click More Info.

Tab 1

1011 YTD [Quantity]

1070 Prior YTD [Quantity]

433 R6M Classic [Quantity]

276 Same Month Last Year [Quantity]

ASOFDATE: 2018-12-12

Quantity (Count)

201701 201702 201703 201704 201705 201706 201707 201708 201709 201710 201711 201801 201802 201803 201804 201805 201806 201807 201808 201809 201810 201811 201812

Find

- 201701
- 201702
- 201703
- 201704
- 201705
- 201706
- 201707
- 201708

Next steps: **Join our Community!**

- Access our new client webinar series
- Join discussion forums with peers
- Stay up to date on new product features
- Help steer product roadmaps

ibm.biz/bizanalyticscommunity

