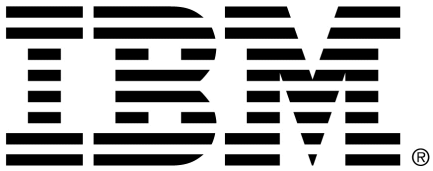


Guideline

Creating Access Tables with IBM Cognos TM1

Product(s): IBM Cognos TM1

Area of Interest: Security

Business Analytics software

Copyright and Trademarks

This document is maintained by the IBM Business Analytics Proven Practices team. You can send comments, suggestions, and additions to cscogpp@ca.ibm.com.

Table of Contents

Introduction

Purpose of Document

This document will describe how to set up security access tables for IBM Cognos TM1.

Applicability

IBM Cognos TM1 10.2 and IBM Cognos TM1 10.2.2

Exclusions and Exceptions

There are no known exceptions and exclusions at the time this document was created.

Assumptions

Users should be familiar with the functionality of IBM Cognos TM1. The document will not discuss business rules, cubes and/or dimensionality as it will assume users are familiar with IBM Cognos TM1 functionality. The target audience is IBM Cognos TM1 administrators as they are individuals that will be working with IBM Cognos TM1 application web.

The document target audience is for consultants that have familiarity with Cognos Planning and TM1. Cognos Planning terms are express in the document as way to overlay the information with the capabilities of TM1 Application Web.

Setting up Access tables with IBM TM1 Application Web

Background

This document describes the way in which security and data access controls can be applied in an IBM Cognos TM1 Application. The document references Access Table functionality from IBM Cognos Planning and describes how IBM Cognos TM1 techniques can be used to implement comparable behaviour to access tables.

Access tables

Access tables in IBM Cognos Planning allow the modeller to specify a data access level for members of one dimension, or combinations of members of more than one dimension. Customers are advised not to create access tables that combine many dimensions. A common requirement is to control access to members of one dimension (such as Accounts) in addition to the eList (approval hierarchy) dimension (for example, Cost Centres or Businesses).

Access tables in Cognos Planning actually perform three functions which behave somewhat independently in TM1.

- **Managing data sparsity** - The underlying EP (Enterprise Planning) Analyst model may infer a large pre-cut down slice but the package deployed to EP Contributor end users for a particular eList node will be smaller if access tables are defined appropriately. This is important where access tables use NO DATA to define only the feasible combinations of two or more dimensions – for instance, the relevant Accounts for a particular Business, or the relevant Products that may be sold to a given Customer.
- **Setting data security** - In EP Contributor, access tables are also the vehicle used to set data security – for instance whether a given Account Code is hidden or is presented read-only. Note that in IBM Cognos Planning, any user opening a given eList item will see the same data since the security is a function of the eList, whereas in IBM Cognos TM1 security is driven by Groups. Therefore it is possible to model some situations in IBM Cognos TM1 that wouldn't be possible in IBM Cognos Planning – for example, it is possible to define a Cost Centre expense model that shows a Salaries cube to members of a Managers group but hides it from members of a Planners group who open the same eList node.
- **Managing unfeasible combinations** - This is a matter of presentation for the end user - if an access table has been defined, the IBM Cognos Planning rich client will suppress away intersections of the cube that the user should not see. For example, if the Forecast member of a Versions dimension is defined as HIDDEN in a single dimension access table, then the client will not show the Forecast member in any cube using the Versions dimension. A more complex situation arises if the access table uses two or more dimensions, such as Accounts and Businesses. If the end user pivots a view of their cube such that an Account (on rows) for a given Business (shown on titles) would be HIDDEN or NO DATA in an Access Table, then the EP Client will suppress the display of that Account.

Table 1 – Comparison of security options between IBM Cognos Planning and IBM Cognos TM1

Function performed by Access Tables	Implementation in IBM Cognos Planning	Equivalent technique in IBM Cognos TM1
Managing data sparsity	NO DATA access tables in conjunction with Cut Down Models (EP <10.1) or Access Blocks (EP 10.1)	Sparse consolidation algorithm and use of Feeders
Setting data security	HIDDEN, READ, WRITE access table entries define what cells the user of a given eList node can see or write to.	Cube, Element, Cell Security, or Rules applied to the data entry cube. Security managed by Group memberships.
Managing unfeasible combinations	NO DATA and HIDDEN access tables enforced and unfeasible combinations are suppressed in the EP Client.	Single dimension access tables will behave as for element security. The client will suppress dimension members that the user has no rights to

Function performed by Access Tables	Implementation in IBM Cognos Planning	Equivalent technique in IBM Cognos TM1
		see. For multi-dimensional access tables, there is no directly comparable concept. Techniques are available in TM1 to help suppress unfeasible combinations.

General principles for setting TM1 security

The most general principle is to keep the security implementation as simple and straightforward as possible. Start by thinking of security at the most coarse level and only use finer-grained techniques where necessary. Similarly, in a TM1 Application, consider whether security needs to be set for specific groups or whether the **}tp_Everyone** group (for native secured models) or the **Cognos\Everyone** group (for Cognos Access Manager (CAM) secured models) will suffice.

- **Cube Security** - Can I achieve my requirements by allowing some users access to a particular cube, but not others? For example, in an Expense Planning application, it is possible to have a Managers group able to write to a Salaries cube but not allow a Planners group any access to the same cube – even if both groups of users have access to the same item in the approval hierarchy of the application.
- **Element Security** - Many common modelling requirements can be met with element security. For example, ensuring that Actuals are read-only for all users, whereas Budget is writeable and ensuring that when a forecast is rolled forward, Q1 can be made read-only, and Q2, Q3 and Q4 remain writeable. Another advantage of using element security is that when it is set to NONE (or left blank), TM1 clients such as IBM Cognos Insight will strip away that element from a view. This is similar to the behaviour of a Cognos Planning access table using a HIDDEN setting. Note that the NONE setting in TM1 should not be confused with NO DATA in Cognos Planning – where an element (or cell) has NONE access in TM1, it is still possible for data to reside at that location whereas in Cognos Planning, the use of NO DATA means that data can't exist there, even if imported from an external source.
- **Cell Security** - The driver for using cell security is where the modeller needs to control access to elements of two or more dimensions in combination. Note that if the requirement is 1) to be able to control feasible combinations of dimensions (such as Account and Business or Products and Customers) and 2) this constraint should apply to **all** users of the Application, then it is possible to use Rules in the data entry cube rather than using cell security. However, if the business requirement is to define different security levels (WRITE, READ and NONE) for different Groups across two or more dimensions, this can only be achieved with cell security. Note that where cell security is defined for a particular cell, it has the effect of overriding any other security (e.g. element security). However, where cell security is not explicitly set for a given cell, element security will still propagate up and take effect.

These concepts are illustrated below with a simple case study.

Security case study

Consider a simple Plan Entry cube with dimensions of Accounts, Quarters, Businesses and Versions (Figure 1):

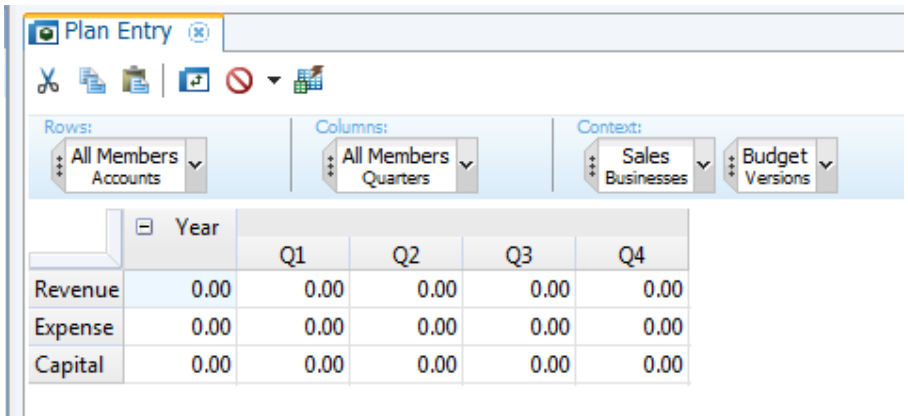


Figure 1 – cube layout for plan entry

These are the members of all the dimensions used in the Accounts, Quarters, Businesses and Versions cubes.

Table 2 – Cube Dimensions and elements

Dimensions	Elements
Accounts	• Revenue • Expenses • Capital
Year	• Q1 • Q2 • Q3 • Q4
Businesses	• Sales • Back Office • Finance • Manufacturing
Versions	• Actual • Budget • forecast

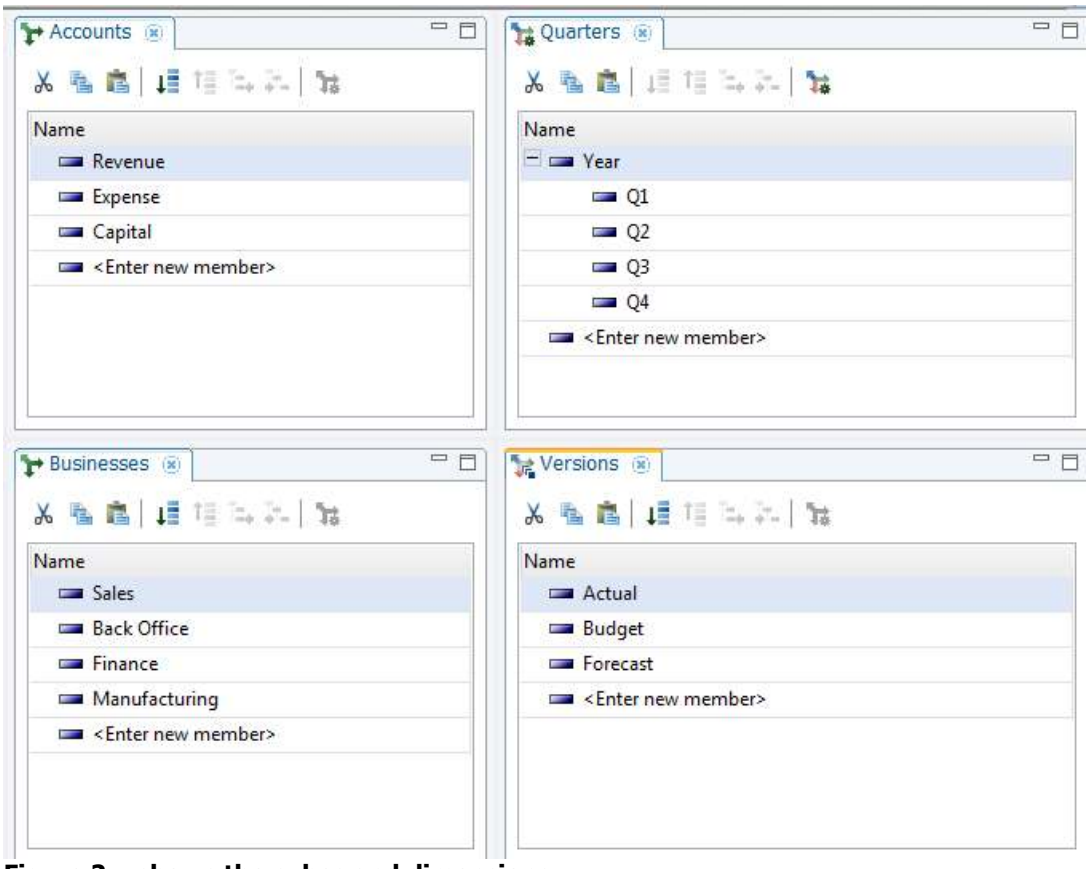


Figure 2 – shows the cubes and dimensions

Note that in Figure 2, the user has logged into TM1 Application web as an Administrator and so the user will not directly see some of the security constraints taking effect in the when views are open by the user – it is important to recall that cube, element and cell security are invisible to an Administrator, who is permitted to make data changes to cube data without being blocked by these forms of security.

Element Security on Versions and Quarters

Our first requirement is to make the Actual version read-only. TM1 administrators can request the dialog to set access permissions for dimension elements. TM1 administrators can select Configure Security\Set Access Permissions for\Elements as illustrated in Figure 3.

Hint: There are also options to set security for dimensions, cubes, processes and chores.

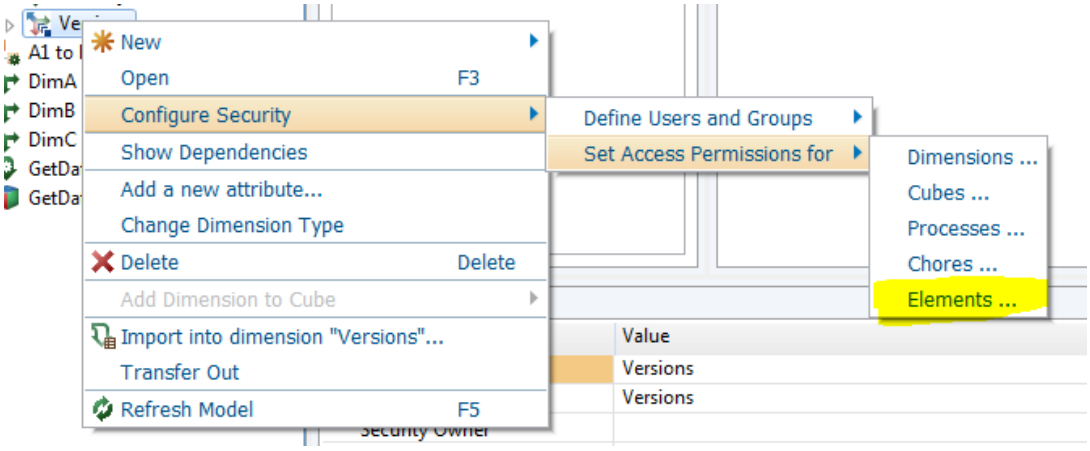


Figure 3 – shows setting access permission for elements

As shown in Figure 4, in the resulting security editor TM1 Administrator can set READ access for Actual and WRITE access for Budget. TM1 Administrator will leave the Forecast line blank. Note that we apply these settings to the }tp_Everyone Group as we want this to apply to all users of the TM1 Application. When approval hierarchy rights for a TM1 Application are saved, every member of any Group assigned rights to the Application will also be added to the }tp_Everyone Group.

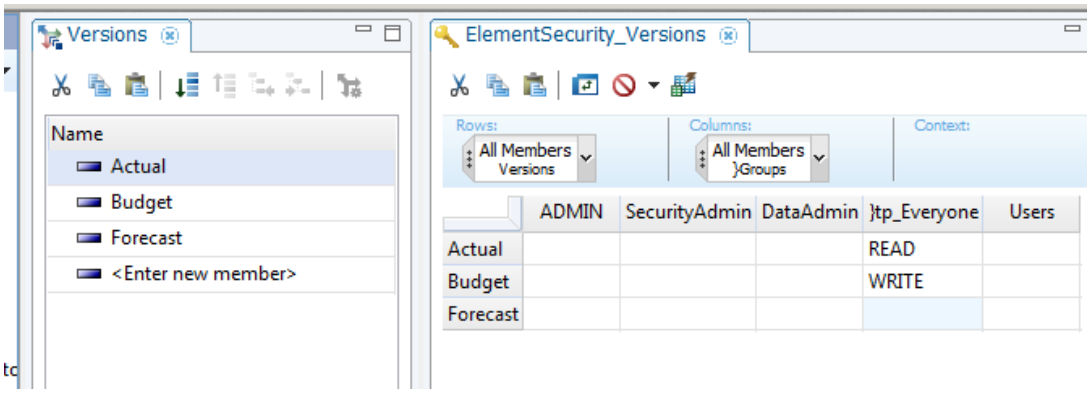


Figure 4 –Security permission options

When a regular TM1 user of the application opens the Cognos Insight client (in this case for the Sales approval hierarchy node), the TM1 user will see this view in figure 4 of the Plan Entry cube. Note that the TM1 user's view of an Approval or Responsibility application will be greyed out until the TM1 user takes ownership.

TM1 user will see that the Actual slice is greyed out because it is read-only, though we can see the underlying values of zero. TM1 user is able to key to the Budget slice (which TM1 administrator had marked as WRITE) but the Forecast slice is not shown at all. The blank entry in the element security editor for Versions is interpreted as NONE, and the client suppresses away the Forecast slice.

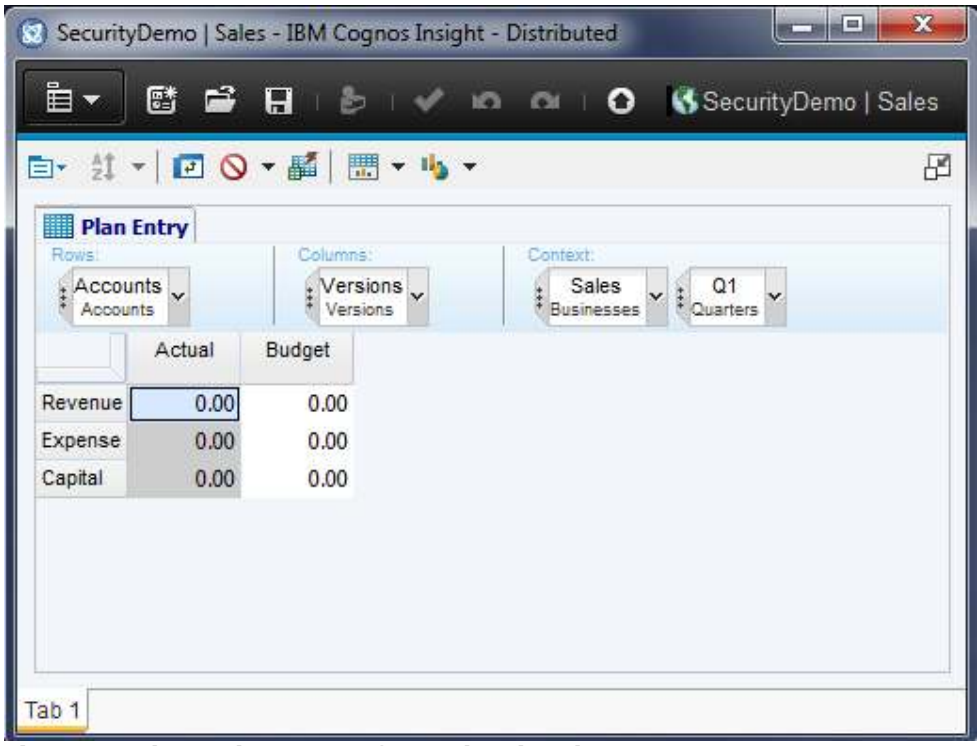


Figure 5 – shows the status of Actual and Budget

Now return as the TM1 Administrator to TM1 Application Web and set element security on the quarter's dimension. Again, see the TM1 Administrator has applied different security access to the }tp_Everyone Group by quarter.

Table 3 – shows security access by dimension and security group

Dimension	Security Groups	Security access
Year	Admin	
Year	SecurityAdmin	
Year	DataAdmin	
Year	}tp_Everyone	Write
Year	Users	
Q1	Admin	
Q1	SecurityAdmin	
Q1	DataAdmin	

Q1	}tp_Everyone	Read
Q1	Users	
Q2	Admin	
Q2	SecurityAdmin	
Q2	DataAdmin	
Q2	}tp_Everyone	Write
Q2	Users	
Q3	Admin	
Q3	SecurityAdmin	
Q3	DataAdmin	
Q3	}tp_Everyone	Write
Q3	Users	
Q4	Admin	
Q4	SecurityAdmin	
Q4	DataAdmin	
Q4	}tp_Everyone	Write
Q4	Users	

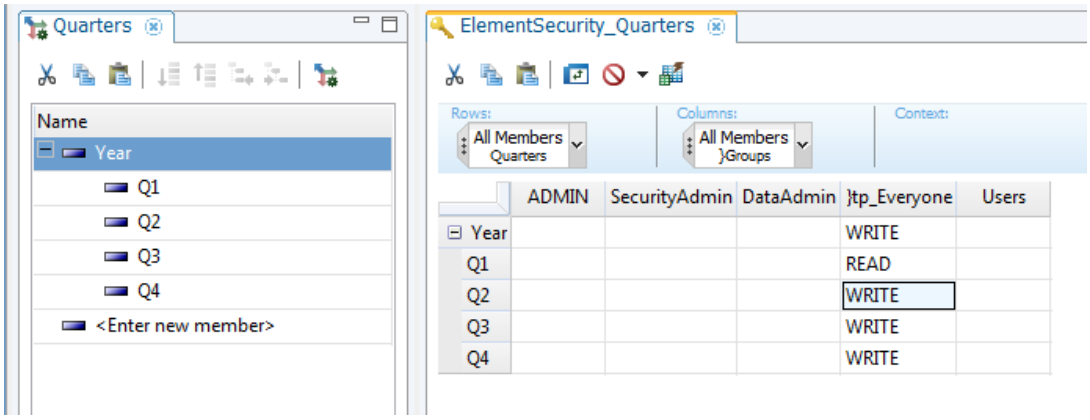


Figure 6 – shows setting of security options by Year dimension

Returning to Cognos Insight as a regular, non-Admin user, we see in Figure 7 that Q1 is greyed out since it has been set to read-only.

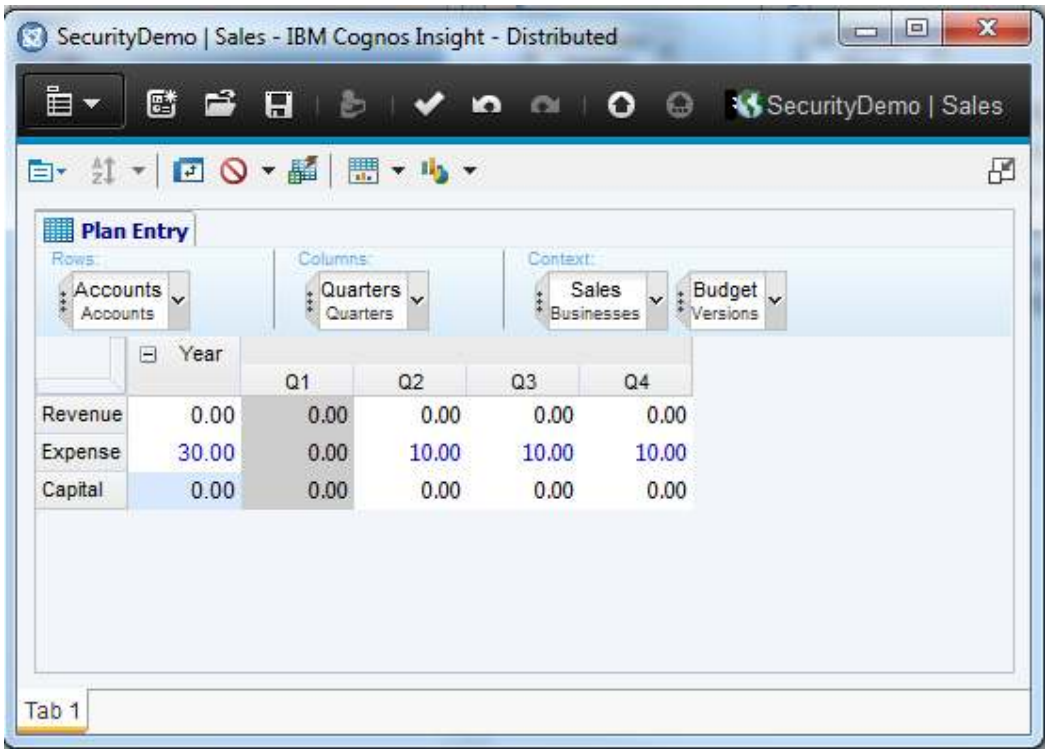


Figure 7 – shows Q1 in read only mode for security

So Q1 is now read-only but as illustrated in Figure 8, TM1 user will still able to spread data from the Year into the three open, writeable Quarters for the Budget slice. If the TM1 user stack Versions onto columns the TM1 user can then see that the whole of the Actual slice is still read-only and the whole Forecast slice is still not shown at all.

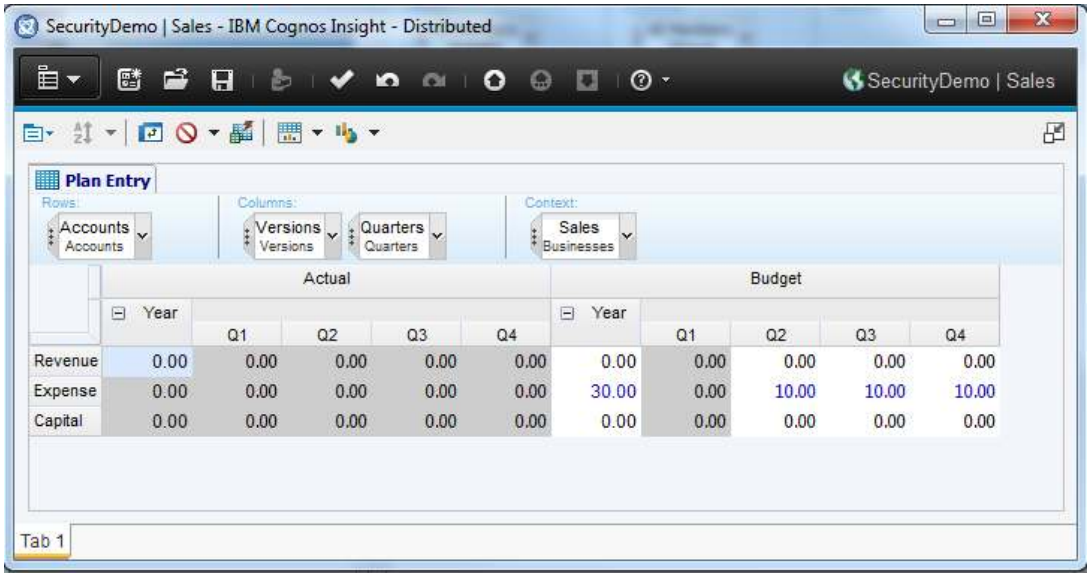


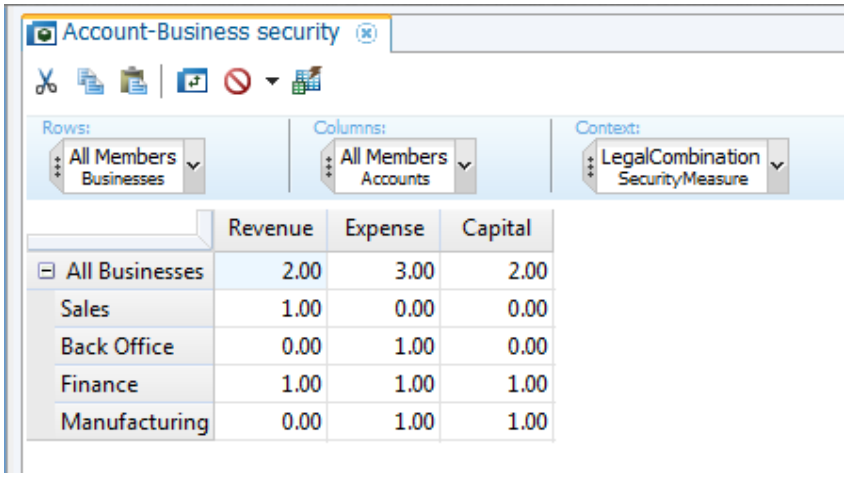
Figure 8 – shows read and write security settings for Actual and Budget

So TM1 administrators can see that element security on Versions and Quarters can be set independently but their effects are complementary. Generally, if TM1 administrator applies' different security rights to the objects that identify a cell of data, TM1 applies the most restrictive security right to the cell (though this is slightly different with cell security, which overrides all other TM1 security and is discussed later). So if a Group had READ security to an element in one dimension and WRITE security to an element in another dimension, then at the intersection of those dimensions, members of that Group would have READ access.

Permitting data entry for some Account and Business combinations

The next TM1 administrator will be is to ensure that TM1 users can only key data for certain valid combinations of Account and Business. There are different ways of achieving this requirement but TM1 users relies on TM1 defining what the valid combinations are. To start, TM1 administrator will define a new cube, **Account-Business security**, where TM1 administrator identifies what combinations of Account and Business are legal.

As shown in Figure 8, TM1 users can only plan for Revenue in the Sales business but the Manufacturing business plans for both Expense and Capital. The zero value intersections represent illegal combinations – TM1 administrators do not want users to plan for Revenue or Capital accounts for the back office business.



	Revenue	Expense	Capital
All Businesses	2.00	3.00	2.00
Sales	1.00	0.00	0.00
Back Office	0.00	1.00	0.00
Finance	1.00	1.00	1.00
Manufacturing	0.00	1.00	1.00

Figure 8 – shows the security setting for account-business security

Enforcing data access with Rules

The first way of enforcing this constraint is to use a rule in the Plan Entry cube:

```
[] = IF ( DB ('Account-Business security', !Accounts,
!Businesses,'LegalCombination')>0,CONTINUE,0);
```

This rule effectively states that if the **LegalCombination** flag in the **Account-Business security** cube is non-zero, processing of rules should continue, else if the **LegalCombination** flag is zero, the Plan Entry cube should display a zero value. The presence of the CONTINUE statement means that if no other rules lower down the rule file impact a given cell of the cube, it will remain open for users to key data.

Note that that figure 9 shows a view of the Plan Entry cube in TM1 Application Web, where TM1 user is logged in as the TM1 Administrator. TM1 administrator can see that for the Back Office as the only area that data can be keyed into the Expense account but the Revenue and Capital accounts are greyed out. This is because they are rule-derived – as even as a TM1 Administrator cannot key data into plan entry. Also note that TM1 Administrator can write to the Expense account for Q1. This is because the element security that makes Q1 read-only has no effect for TM1 user as the TM1 Administrator.

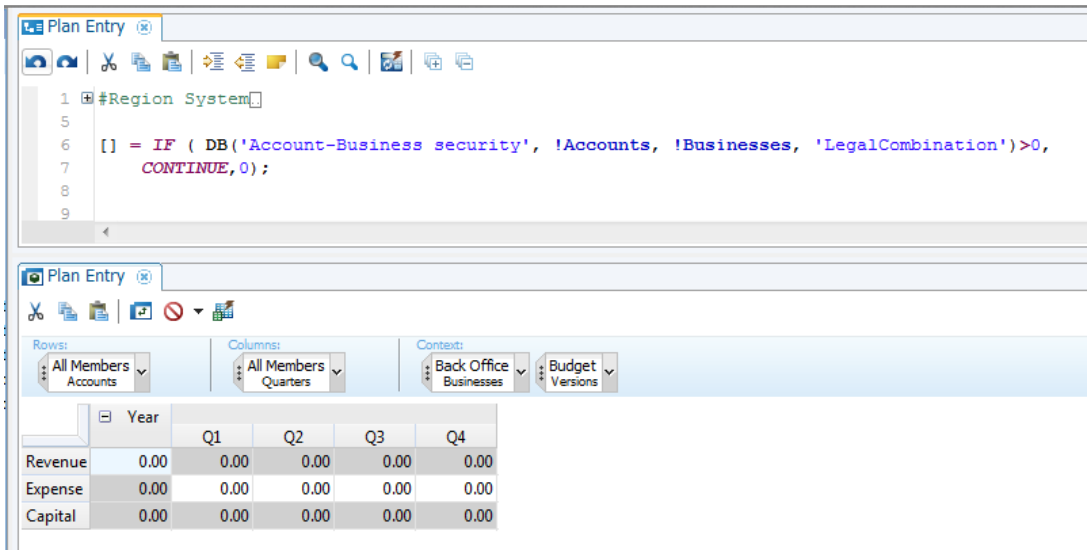


Figure 9 – shows security settings for revenue, expense and capital

When TM1 Administrator returns to Cognos Insight as a regular TM1 user, this is what the TM1 user will now see in figure 10 if the TM1 user open the Back Office node and take ownership

Note that the Revenue and Capital lines are still displayed, even though our business logic has forced their values to zero for the Back Office. TM1 user could apply zero suppression to remove them, though this would also suppress the Expense line as well unless TM1 user has non-zero data entered for Expenses. If TM1 user wished, TM1 user could use the **Blank if zero** format option to have the zero values for Revenue and Capital displayed as blank cells, even though they actually contain a zero value. There is another advanced technique that will be discussed later in the document for suppressing whole rows of data that represent these unfeasible dimension combinations.

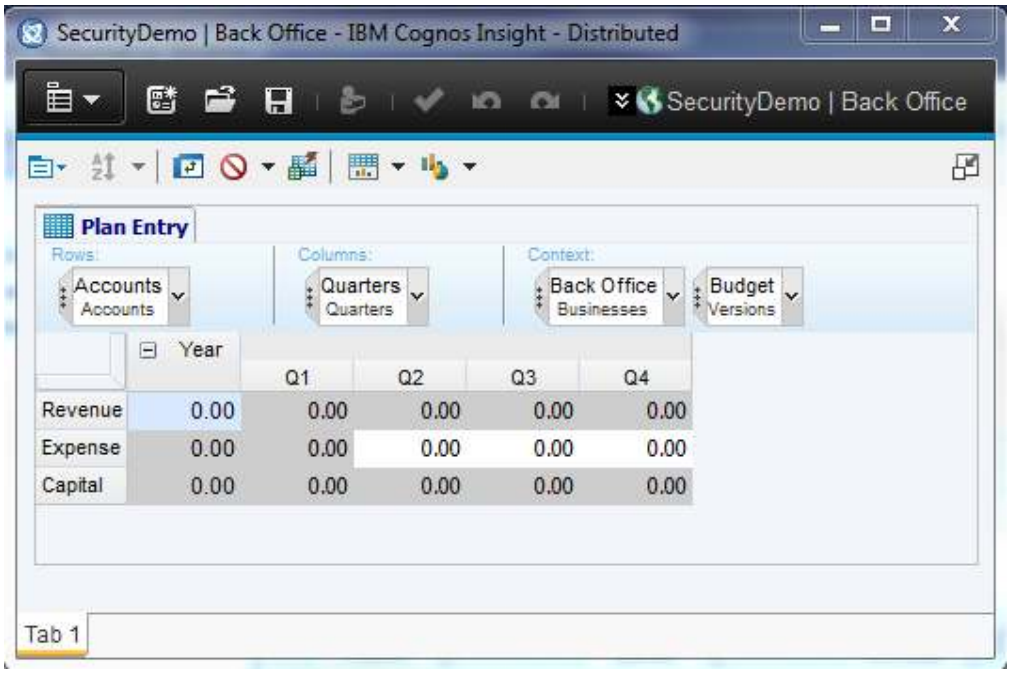


Figure 10 – shows cube security in Cognos Insight

Figure 11 shows the TM1 user entering amounts for the expense dimension by quarter using Cognos Insight.

Table 4 – shows the data entry for expense dimension by period

Account Dimension	Period Dimension	Amount
Expense	Year	50.00
Expense	Q1	0.00
Expense	Q2	20.00
Expense	Q3	30.00
Expense	Q4	0.00

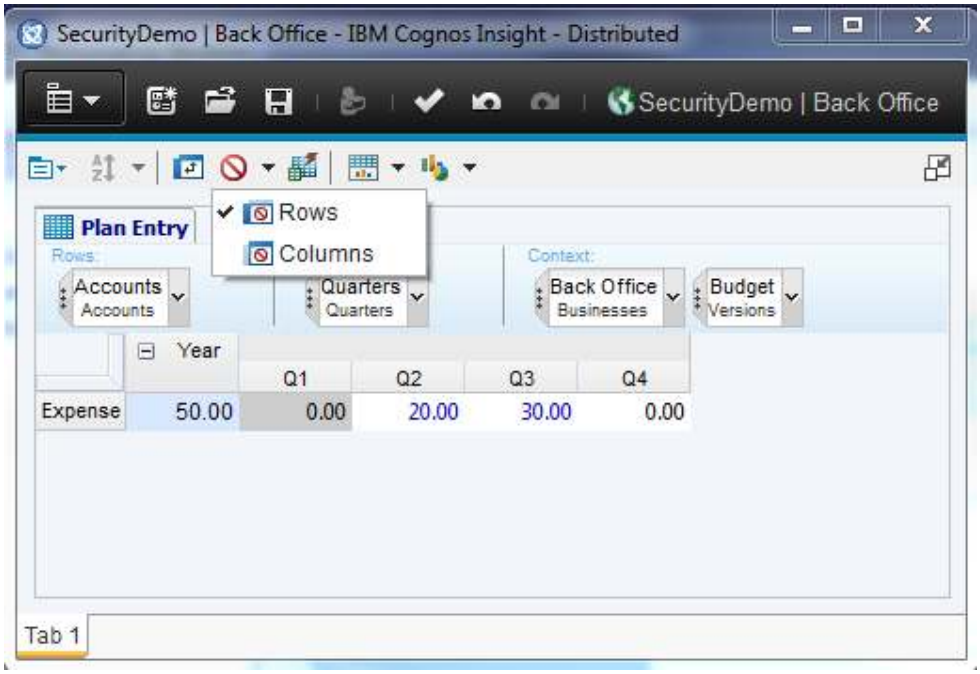


Figure 11 – shows the expense dimension security access

Populating cell security with a Process

Before TM1 Administrators can re-use the Account-Business security cube, TM1 Administrator will need to adjust it to display strings by adding a string element to the **SecurityMeasure** dimension called **SecuritySetting**. As shown in Figure 12, TM administrator can see that the populated the **illegal** combinations, whereas previously when TM1 administrator used Rules in the Plan Entry cube TM1 administrator put a flag value of 1 against the **legal** combinations.

The reason for this is that cell security will override element security where an entry exists for a specific cell in the cell security cube – and this is the behaviour TM1 administrator wants for the cube/dimensionality. In other words, unless we've flagged an Account-Business combination as being illegal using the NONE flag, TM1 Administrator still want any element security on Versions and Quarters to propagate through in the Plan Entry cube.

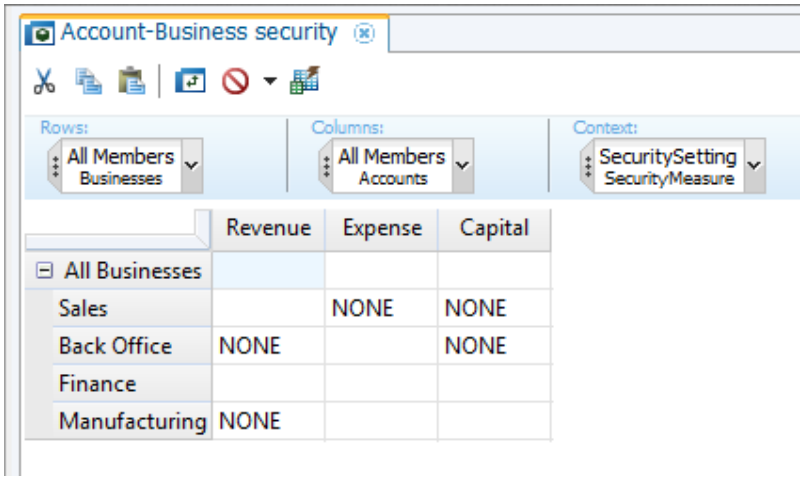


Figure 12 – shows setting of cell based security for the plan cube

Figure 13 shows the moving of security information from the Account-Business security cube to the Plan Entry cell security cube, TM1 Administrator can create a Link but first the TM1 Administrator must **Show Control Objects** so that TM1 administrator can select the cell security cube as the target of the Link. TM1 administrator then can configure the Link as stated in Figure 13 – setting the **Link Implementation Type** to **Process**, and using a manual mapping between **SecurityMeasure** (in the source) and **}Groups** (in the target) and mapping **SecuritySetting** to the **}tp_Everyone** group. TM1 administrator have clicked the **select all** checkboxes for Quarters and Versions.

Note: If TM1 administrator had a requirement to set different security for different groups, TM1 administrator could add a Groups dimension to the Account-Business security cube but TM1 administrator won't be doing that in Figure 13

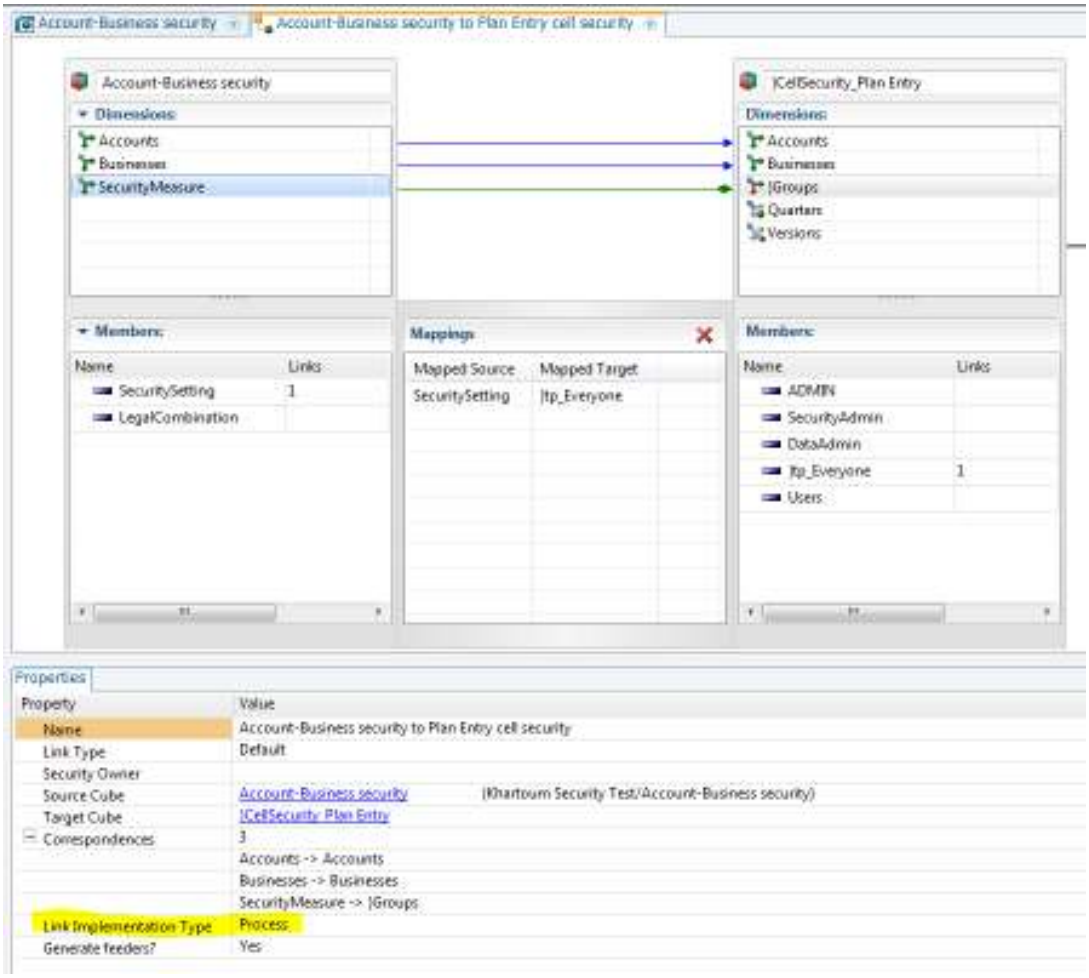


Figure 13 – shows creation of admin link to the control cube for security

TM1 administrator then can save the Link and right-click it in the model tree to Generate **Process** – see figure 14

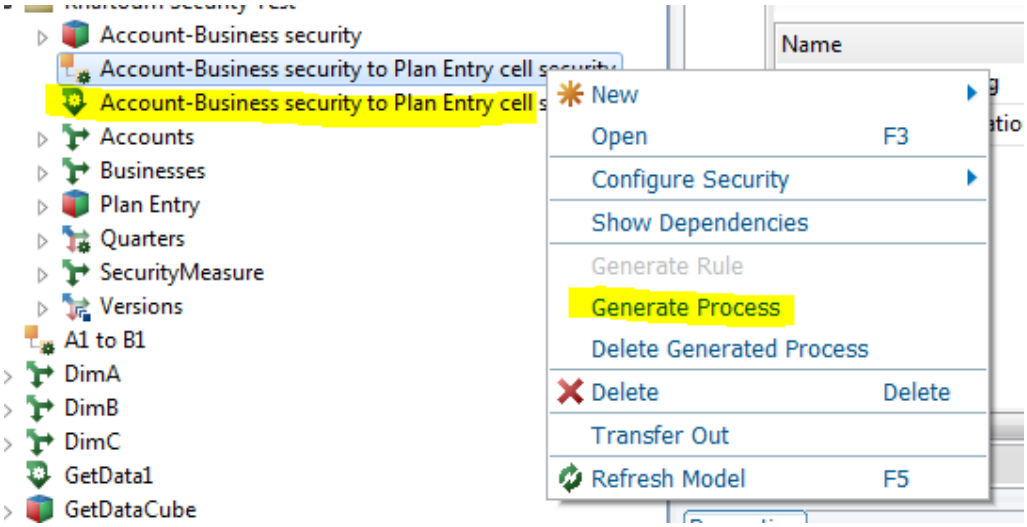


Figure 14 – shows generation of the admin link

TM1 Administrator then can see that a process is created with the same name as the Link. If we right-click this process and select **Execute Process** (leaving the parameter prompts blank), the security information is copied to the cell security cube. If we right-click the Plan Entry cube and request to see the cell security editor again, TM1 administrator can verify the result – see figure 15

Table 5 – Security setting by cube

Cube	Dimension	Dimension	Security
Account-Business security	Sales	Revenue	
Account-Business security	Sales	Expense	None
Account-Business security	Sales	Capital	None
Account-Business security	Back Office	Revenue	None
Account-Business security	Back Office	Expense	
Account-Business security	Back Office	Capital	None
Account-Business security	Finance	Revenue	
Account-Business security	Finance	Expense	
Account-Business security	Finance	Capital	
Account-Business security	Manufacturing	Revenue	None
Account-Business security	Manufacturing	Expense	
Account-Business security	Manufacturing	Capital	
Plan Entry	Sales	Revenue	
Plan Entry	Sales	Expense	None
Plan Entry	Sales	Capital	None
Plan Entry	Back Office	Revenue	None
Plan Entry	Back Office	Expense	
Plan Entry	Back Office	Capital	None
Plan Entry	Finance	Revenue	
Plan Entry	Finance	Expense	
Plan Entry	Finance	Capital	
Plan Entry	Manufacturing	Revenue	None

Plan entry	Manufacturing	Expense	
Plan Entry	Manufacturing	Capital	

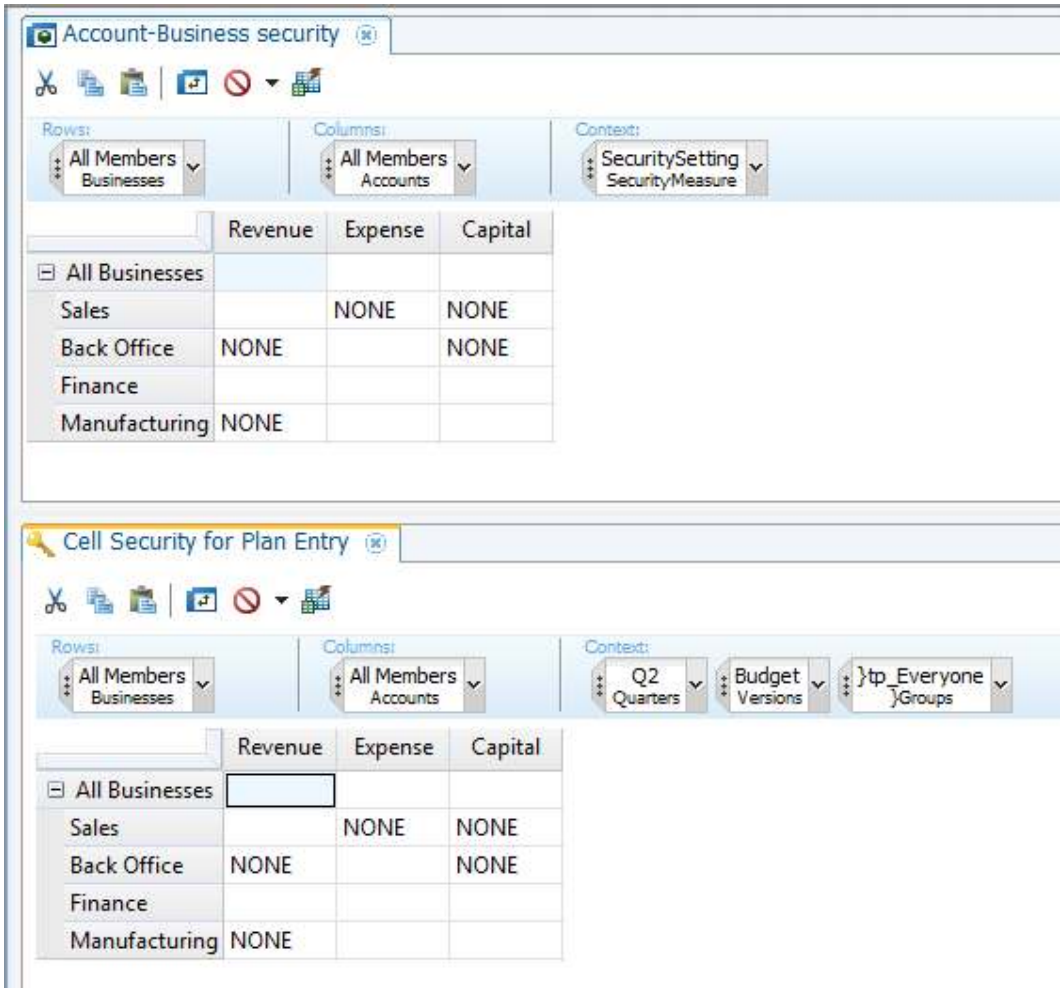


Figure 15 – The cell security has been correctly applied to Plan Entry cube

If TM1 Administrator re-deploy the application and then open the Back Office node as a regular TM1 user in Cognos Insight, the TM1 user will see the result (Figure 19). The rows for Revenue and Capital are shown as blank because we defined security for those Accounts as NONE in the Back Office. Additionally, TM1 user can see that the Expense account is read-only for Q1 because element security on the Quarters dimension takes effect for that cell.

If the TM1 user re-orient the view to stack Versions on columns, TM1 user will see in figure 16 that the whole of the Actual slice remains either read-only (in the case of the Expense account) or we have NONE access (for the Revenue and Capital accounts, shown by the blank, grey cells

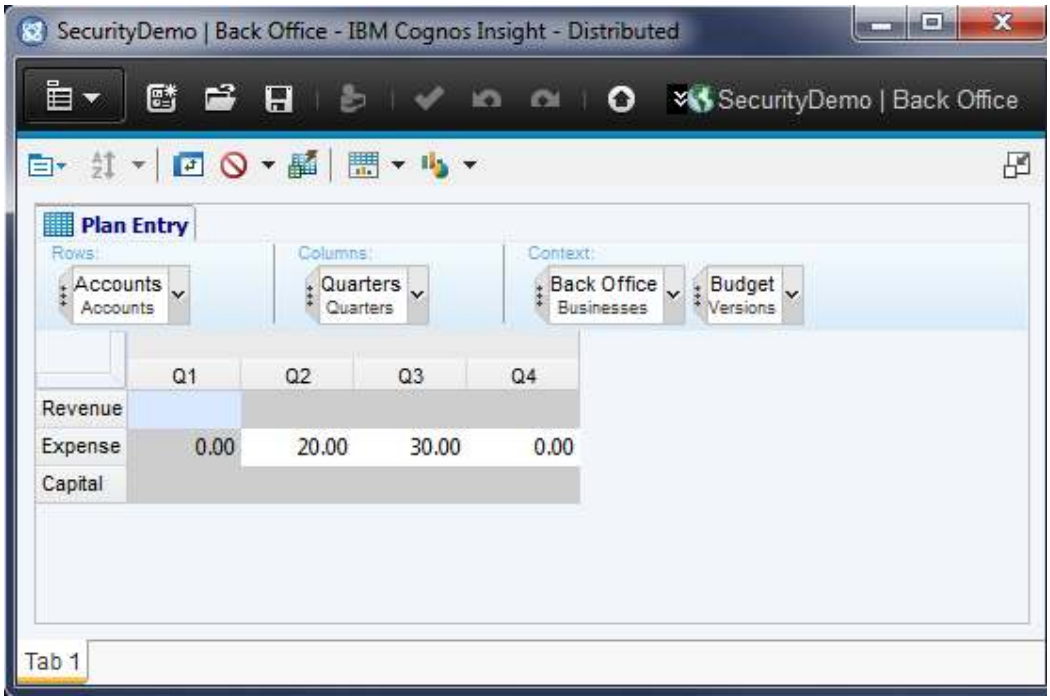
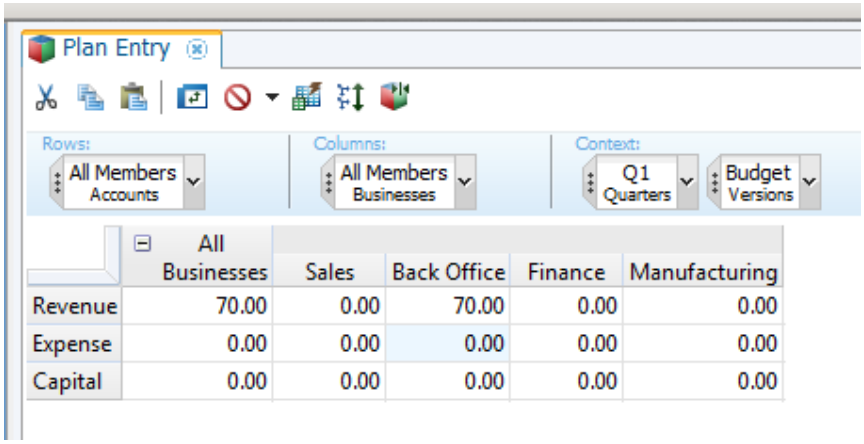


Figure 16 – shows security for the Plan Entry cube applied at the cell level

Note that the Revenue and Capital lines are not suppressed away in figure 17, even though TM1 user has no access to data for Revenue and Capital. The TM1 client will remove and suppress elements where a TM1 user has NONE access if using element security but if cell security is applied and the TM1 user has no access to a whole row of data in a view, then the row will still be shown.

Note: If zero suppression is switched on, then the rows will be suppressed assuming that the cells to which TM1 have no access do indeed contain a zero value. If any of the Revenue or Capital cells contained non-zero data, an end user would still not be able to see the data but they would observe that switching on zero suppression would not suppress those cells.



	All Businesses	Sales	Back Office	Finance	Manufacturing
Revenue	70.00	0.00	70.00	0.00	0.00
Expense	0.00	0.00	0.00	0.00	0.00
Capital	0.00	0.00	0.00	0.00	0.00

Figure 17 – shows the plan entry by business

In Figure 18, a TM1 Admin user has entered a value for Revenue for the Budget slice for Q2 and Q3, which is preventing the whole Revenue row being suppressed in the TM1 user's Cognos Insight client but as a regular TM1 user, who has no rights to see that value.

Note that this behaviour differs from that seen when TM1 user previously used rules in the Plan Entry cube to control data access. In that case, even an Administrator could not enter data into the Revenue account for the Back Office. If it is important that stray data cannot exist in the illegal combinations of Account and Business (as would be the case for NO DATA access table entries in Cognos Planning) then the approach of using Rules in the Plan Entry cube would be preferred.

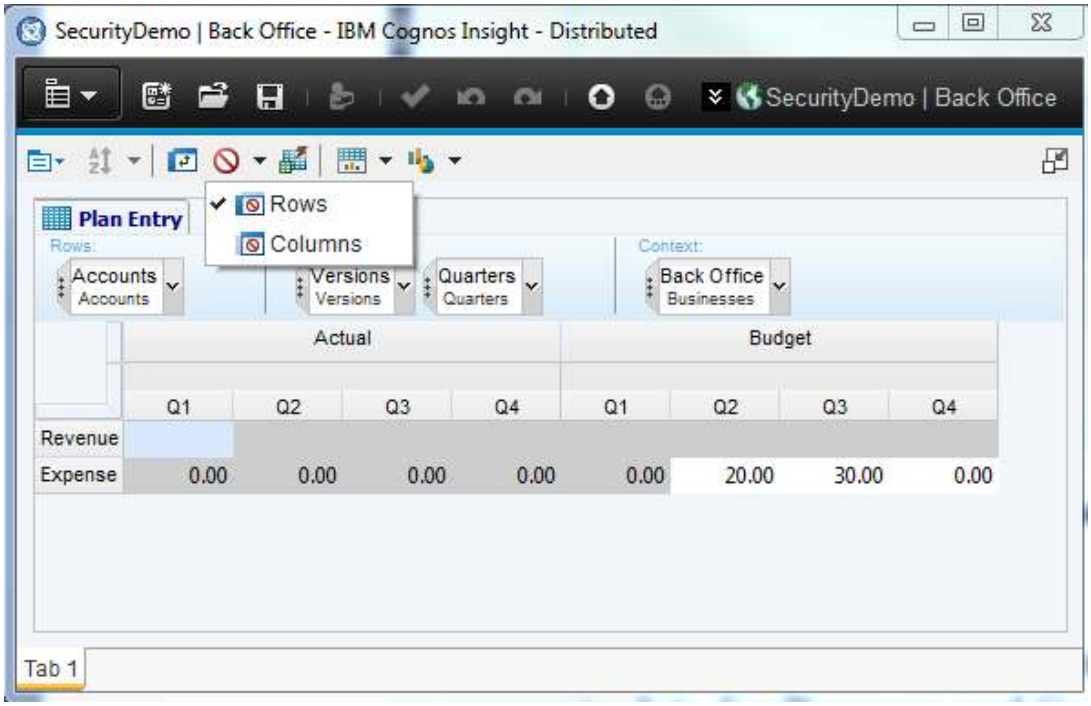


Figure 18 – shows security settings in Cognos Insight for Back Office

Suppressing unfeasible combinations of Account and Business

It is possible to assist TM1 users by suppressing the Accounts that should not be seen for a given Business. This is done by using a dynamic, MDX-based subset on rows for Accounts that references the information in the Account-Business security cube. The subset **Legal Accounts per Business** is constructed using this MDX expression:

```
{FILTER( { TM1SUBSETALL( [Accounts] ) }, [Account-Business
security].( [Accounts].CurrentMember,
[Businesses].CurrentMember,
[SecurityMeasure].[LegalCombination] ) > 0 ) }
```

This is conceptually similar to the way that TM1 user previously used Rules to reference the Account-Business security cube to write zero values to show the illegal combinations as expressed in figure 19.

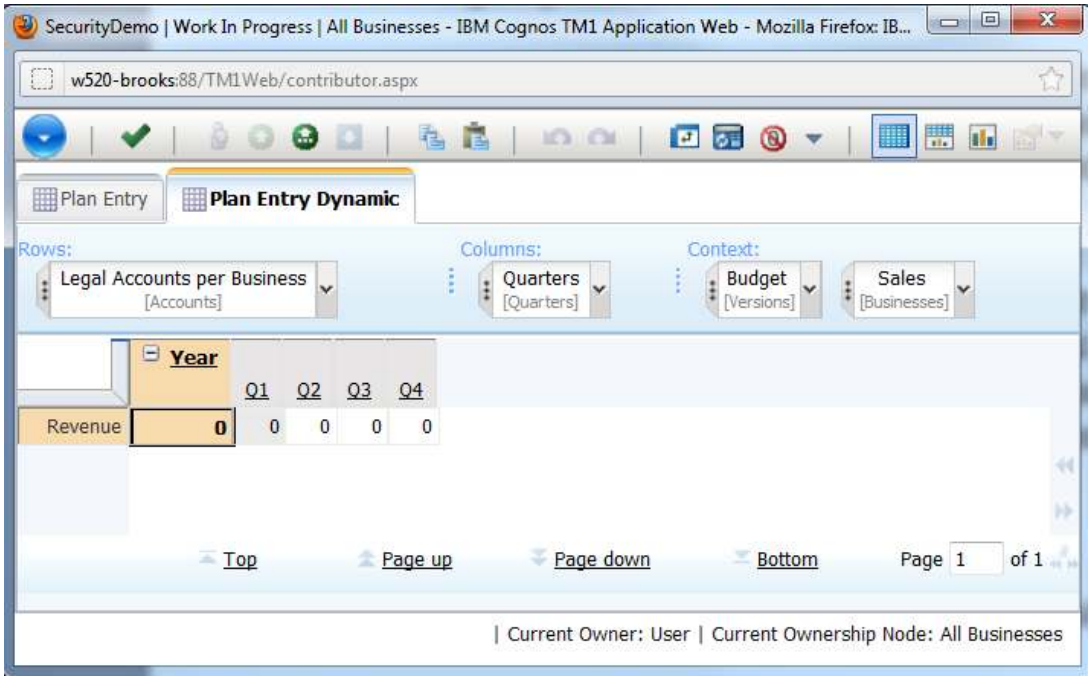


Figure 19 – shows the subset Legal Accounts per Business

Now switch the Businesses title selection to Manufacturing as shown in Figure 20.

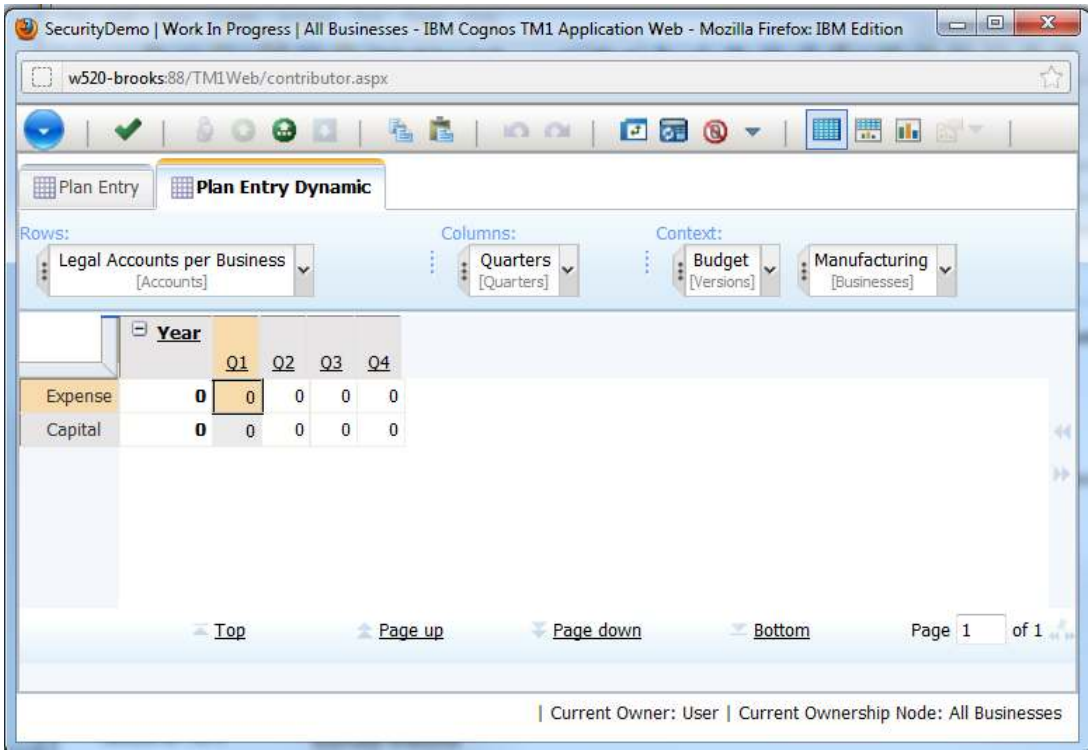


Figure 20 – shows manufacturing security settings

With this simple `FILTER()` expression in the MDX, this technique still works if the view is pivoted or if dimensions are stacked. TM1 users can enter data for Q2, Q3 and Q4 for expense and capital budget for manufacturing context filter as shown in figure 21.

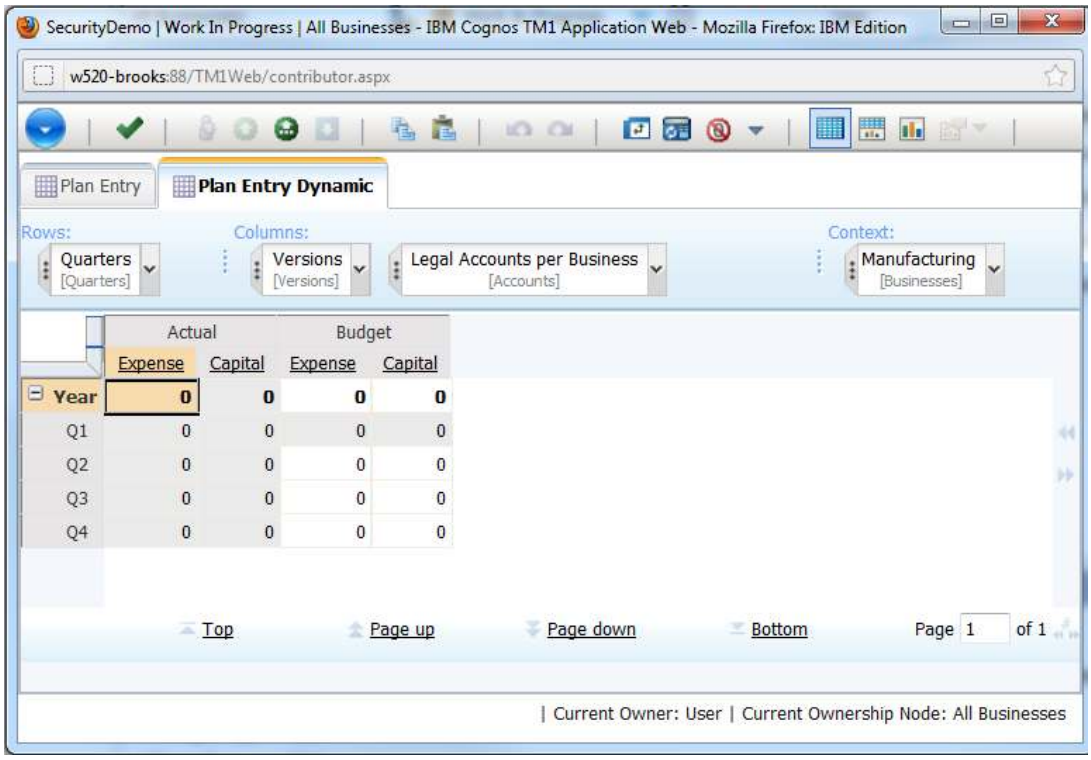


Figure 21 – shows the plan entry for manufacturing

Populating security cubes with Rules

Similar modelling techniques can be used to populate security cubes using Rules rather than manually keying security levels or using a TI Process.

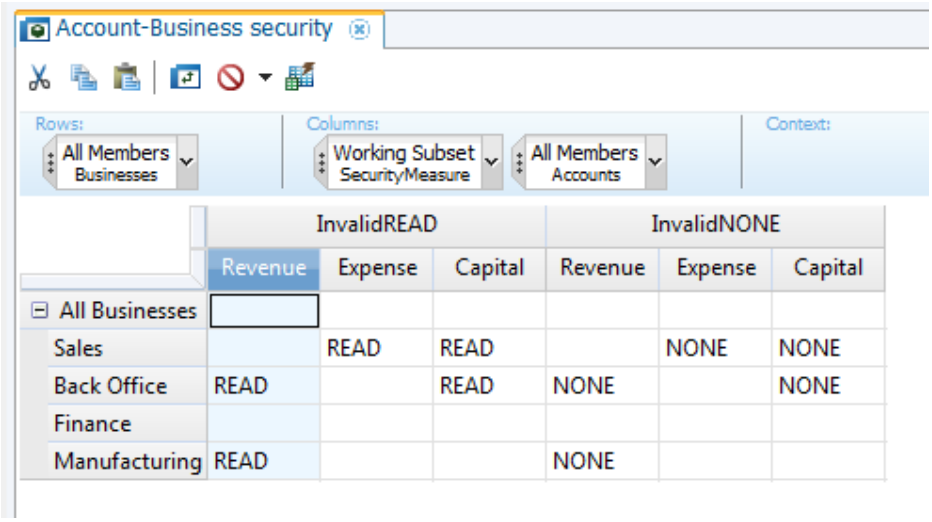
There are two main factors to consider when deciding how security information should be maintained

- **Frequency and nature of update** - Security settings that are maintained by manually keying updates to the user interface or by using TI Processes to populate string values in the corresponding security cubes, will take effect immediately. If rules are used to reference data from a user-created cube in an object security cube (i.e. cube, dimension or element security), then changes to the user-created cube will require a security refresh to take effect. This also applies if rule-derived client-group relationships exist (i.e. if the modeller has written rules against the }ClientGroups cube). A security refresh is a contentious and potentially long-running operation - its effect is to update the security information that is cached in the definition of each object. This restriction does not apply to cell security as cell security information is not cached in the cube to which the cell security applies.

- **Memory footprint** - The }CubeSecurity cube and }ElementSecurity cubes are generally small as they are limited by the number of cubes or elements and by the number of Groups in the TM1 Server. Even if these cubes are densely populated using a TI Process, the memory footprint should be modest. Cell security cubes have greater dimensionality as they reflect all the dimensions of the parent cube plus the }Groups dimension. If the modeller attempts to set very granular security across many dimensions and populates a significant portion of the cell security cube's multi-dimensional space, the cube may become large. Using rules can be an efficient way of populating large areas of a cell security cube without incurring significant memory cost. Note that such rules need not be fed unless the modeller wishes the string data in the target cell security cube to display correctly in a zero-suppressed view. If such a rule were fed, this may also consume large amounts of memory, negating much of the benefit of using Rules rather than a TI Process.

Populating the Plan Entry cell security cube with rules

Let's return to our previous example, where we wished to use cell security to control the valid combinations of Accounts and Businesses. Imagine that we now have two classes of users – those who are not permitted to WRITE to certain Account-Business combinations but can still READ the data and those who are not permitted to see the data at all – users who must have NONE access. To achieve this, we will create two new members in the SecurityMeasure dimension in the Account-Business security cube, named **InvalidREAD** and **InvalidNONE**:



	InvalidREAD			InvalidNONE		
	Revenue	Expense	Capital	Revenue	Expense	Capital
All Businesses						
Sales		READ	READ		NONE	NONE
Back Office	READ		READ	NONE		NONE
Finance						
Manufacturing	READ			NONE		

Figure 22 – shows security settings with rules

TM1 user can now return to the Link that was created earlier and edit it so that the InvalidREAD and InvalidNONE slices are mapped to mapped source to targets for both managers and users as shown in Figure 23) . Note that we can no longer use the }tp_Everyone Group as we're looking for finer control.

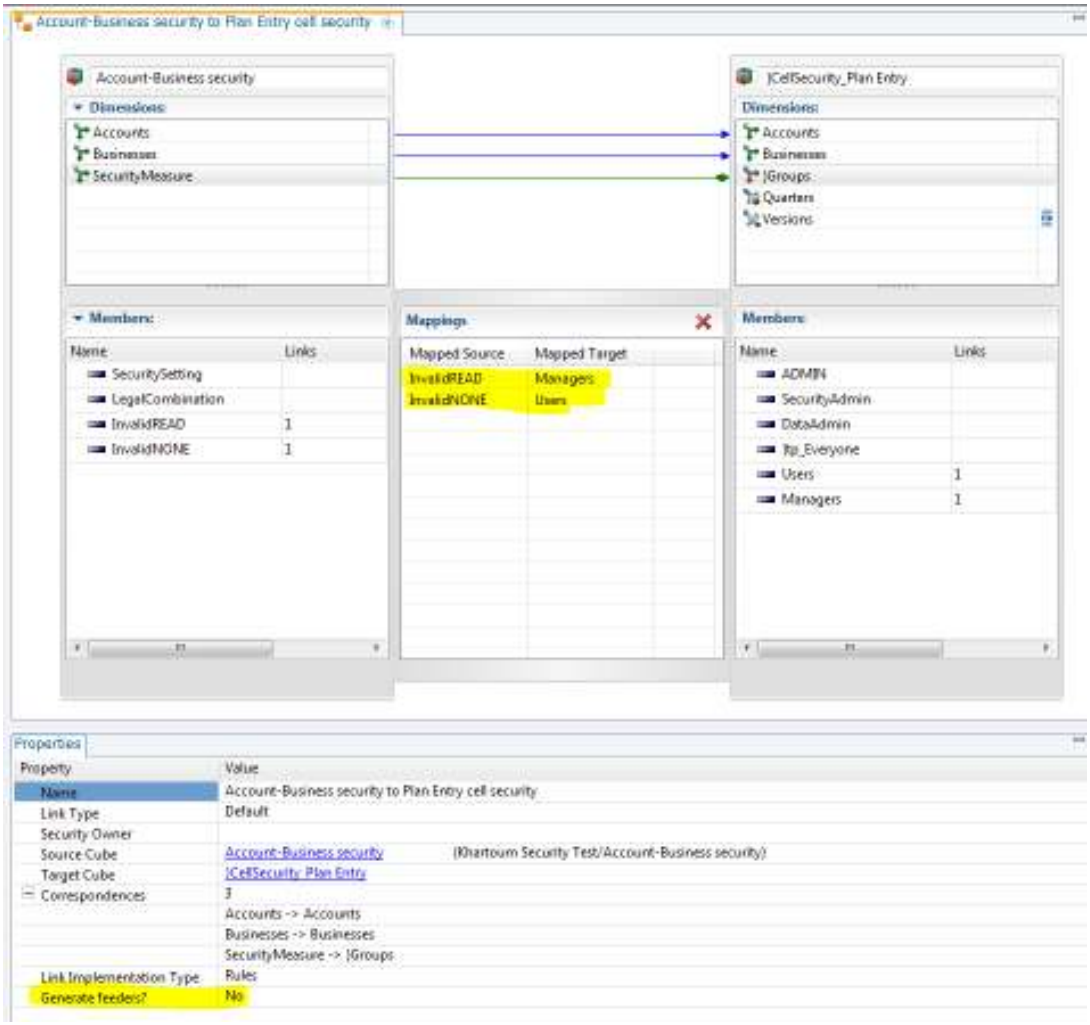


Figure 23 – shows the creation on new link for invalid security options

If TM1 user re-deploy and open the Application as a TM User or a Manager in Cognos Insight, TM1 user will see in Figure 24. In figure 24 users are able to enter amounts for expense for Q2, Q3 and Q4.

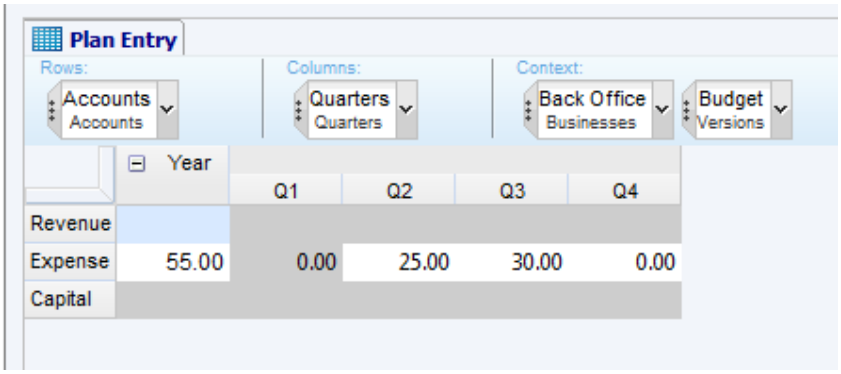


Figure 24 – shows plan entry for expense dimension

When logged in as a TM1 user the behaviour is as before when TM1 user used a Process to populate cell security with the NONE values – user has no rights to see Revenue or Capital data for the Back Office. If TM1 user log in as Manager, a TM1 user can now see that some data does exist for Revenue for the Back Office but TM1 user is only permitted to read those cells, not to write to them. Note that in both cases, TM1 user can still write to the Expense line and Q1 is still read-only owing to the READ element security for }tp_Everyone set on Q1.

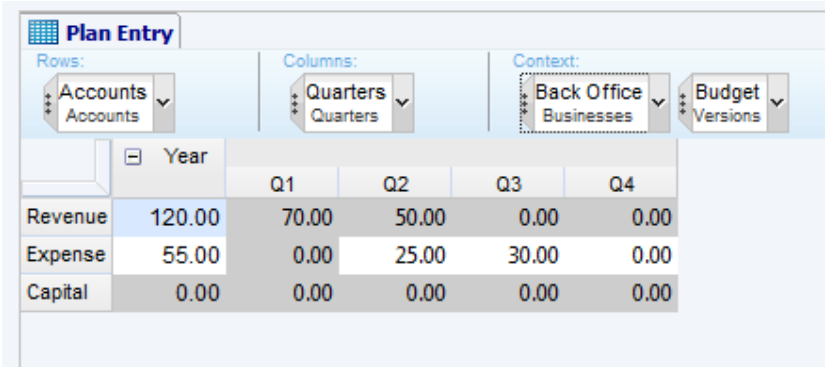


Figure 25 – shows the plan entry security options for back office

If a TM1 User were to show control objects, browse to the }CellSecurity_Plan Entry cube and open its rule string, TM1 User can review the following link rule as part of the security update:

```
1#Region System
2FEEDSTRING;
3SKIPCHECK;
4#EndRegion
5
6#Region Link rule: Account-Business security to Plan Entry cell security –
string
7#Source cube: Account-Business security
8#Target cube }CellSecurity_Plan Entry
9#Autogenerated LINK STRING UNIQUE CODE
10# ['}Groups:('Managers', 'Users', 'Versions':{'Budget', 'Actual'}] =
S:DB('Account-Business security',
11   !Accounts, !Business, ATTRS('{Groups',  !)Groups,   '}Groups
'}Map_}Link_Account-Business security to Plan Entry cell security'));
12#EndRegion
13
14 FEEDERS;
```

TM1 users can switched the **Generate feeders?** option for this Link to No as leaving it set to Yes in a production-sized model could consume a lot of memory and would offer no benefit other than allowing the modeller to browse a zero-suppressed view of the cell security cube.

Populating element security cubes with rules

If the need arose, the same approach could be used to populate element security cubes using rules. For instance, the modeller may wish to build some logic in a cube of their own and then link the resulting security information into an }ElementSecurity cube. In this case, recall that a Security Refresh operation would be needed if the cube upstream of the }ElementSecurity cube were updated. Since }ElementSecurity cubes are unlikely to become very large, it is often better to update them using a TI Process; as shown earlier, this could be done using a Link whose Implementation Type is set to Process.

Summary of TM1 security options

The following table will discuss the security options IBM Cognos TM1

Table 6: Options to secure data

Technique	Advantages	Disadvantages
Cube security	Allows different users to see	Requires a security refresh

	views of different cubes even if they open the same approval hierarchy item from the workflow page. Will take effect immediately if keyed or TI-driven.	to update if rule-driven. The Cognos Insight customised workspace layout does not yet accommodate for this. Widgets containing data from a cube to which you have no rights will produce an error message.
Element security	Controls access to specific members of a dimension. Will suppress away elements that a user has no rights to see. Will take effect immediately if keyed or TI-driven.	Requires a security refresh to update if rule-driven
Rules in the user cube	Can be used to define unfeasible areas of a cube across more than one dimension and force them to zero. Even an Admin user cannot enter or load data to an unfeasible intersection.	Applies to all users - cannot be Group specific.
Cell security	Can be used to define very granular security across any of the dimensions in the cube. Will let element security bubble up if no cell security defined for a given cell. Changes to cell security will take effect immediately without a security refresh, even if rule-derived.	Can be hard to understand at a glance the net result of cell security interacting with element security. Cell security cube could consume large amounts of memory if densely populated using a TI Process. Does not suppress away unwanted cells as element security does.

Table 7: Options to mask unfeasible combinations

Technique	Advantages	Disadvantages
Element security in conjunction with Group naming syntax	If the names used for Groups have a syntax that matches the approval hierarchy node to which they apply, you can set rule-driven element security on another dimension (e.g. Accounts) and effectively have it	You are constrained by a naming convention or use of an attribute to join the approval hierarchy dimension and the other dimension whose members you wish to control.

	combine with the approval hierarchy (e.g. Businesses). Unfeasible Accounts will be suppressed away by the use of element security.	
Dynamic subsets referencing a lookup cube	Can use the same reference cube that you might use to set cell security or rules in the user cube to ensure consistency. Can suppress away unfeasible combinations even when cell security is used.	Requires the modeller to define an MDX-based subset.