

SAP Analytics Cloud model maintenance

Restoring invalid model data caused by hierarchy conflicts



TABLE OF CONTENTS

DEFINING THE PROBLEM	3
EXAMPLE: REPRODUCING THE PROBLEM.....	4
Setting up conflicting hierarchies.....	4
Booking data to parent nodes in a story.....	5
RESOLVING THE PROBLEM	7
Distributing data to leaf members	7
<i>Preparing a table.....</i>	7
<i>Distributing the value.....</i>	9
Changing a model to prevent further problems	10

www.sap.com/contactsap

© 2017 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies. See <http://www.sap.com/corporate-en/legal/copyright/index.epx> for additional trademark information and notices.



DEFINING THE PROBLEM

This document describes the cause of the following warning message in SAP Analytics Cloud, as well as how to resolve the problem and prevent it from occurring again:

The member <Member name> is a parent node containing transactional data in Model <model name>, which causes problems when planning data is entered in Stories.

When working with planning models in SAP Analytics Cloud, all data must be booked to leaf members. In a table based on a planning model, this rule is enforced by automatically booking any data entered on a non-leaf member to one or more leaf members for each dimension in the table.

When creating or editing a model, you can create multiple hierarchies for a single dimension. During this process, it is possible to create conflicting hierarchies, where a dimension member is a leaf member in one hierarchy and a parent member in another hierarchy.

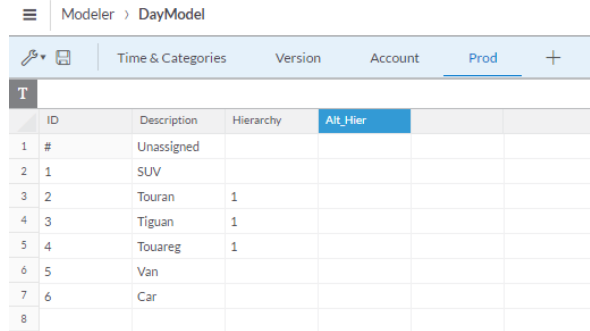
This configuration can lead to an error in a planning model, because it allows users to book a value to a non-leaf member, while its child members all remain unbooked.

In this case, the warning message appears in the Modeler when you make further changes to the model.

EXAMPLE: REPRODUCING THE PROBLEM

Setting up conflicting hierarchies

Consider a planning model with the mandatory Time, Version, and Account dimensions, as well as a generic Prod dimension:



ID	Description	Hierarchy	Alt_Hier
1	Unassigned		
2	SUV		
3	Touran	1	
4	Tiguan	1	
5	Touareg	1	
6	Van		
7	Car		
8			

Image 1: Prod dimension in the sample model

The Prod dimension has two hierarchies. The default hierarchy on the left has two levels, and the SUV members is a parent node. The Alt_Hier hierarchy on the right is flat, and SUV is a leaf node:

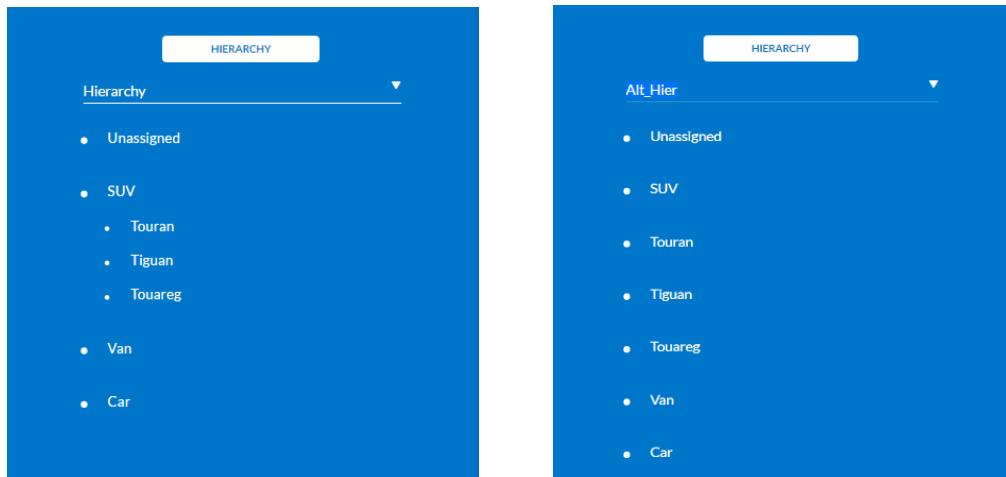


Image 2: Two hierarchies of the Prod dimension

Booking data to parent nodes in a story

In a story, it is possible to book data to the SUV member, which exists as a parent in one hierarchy and a leaf member in the other.

By default, a table based on the model shows the hierarchy where SUV is a parent node:

	A	B	C	D	E	F	G	H
1	DayModel							
2	VERSION	Actuals						
3	VERSION	Actual						
4	PROD	Unassigned	▼ SUV	Touran	Tiguan	Touareg	Van	Car
5	ACCOUNT							
6	▼ All	-	-	-	-	-	-	-
7	Items So..	-	-	-	-	-	-	-
8	Stock	-	-	-	-	-	-	-

Image 3: Table showing the default hierarchy

You can change the hierarchy display for this dimension in the Builder panel by selecting **...** > **Hierarchy** next to the Prod dimension and choosing **Alt_Hier** in the **Set hierarchy** window.

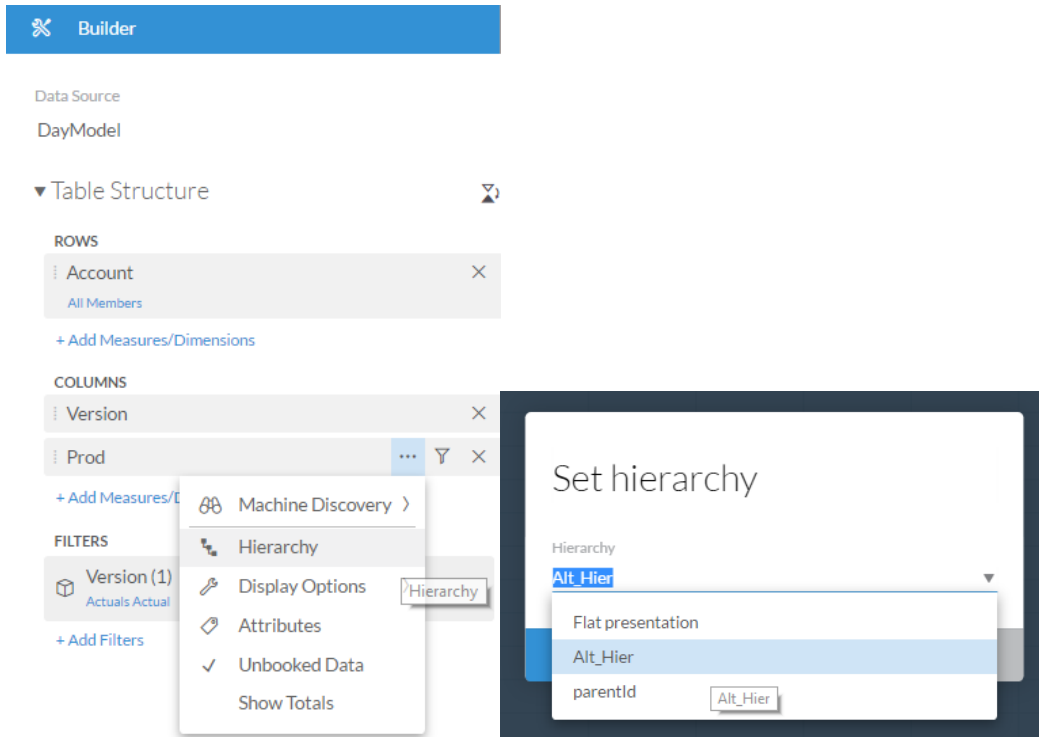


Image 4: Changing the Prod hierarchy

The table now shows the flat hierarchy, where SUV is a leaf member:

	A	B	C	D	E	F	G	H
1	DayModel							
2	VERSION	Actuals						
3	VERSION	Actual						
4	PROD	Unassigned	SUV	Touran	Tiguan	Touareg	Van	Car
5	ACCOUNT							
6	▼ All	-	-	-	-	-	-	-
7	Items So..	-	-	-	-	-	-	-
8	Stock	-	-	-	-	-	-	-

Image 5: Table showing the flat hierarchy

You can now enter data for the SUV member, for example, by selecting cell C7 and typing '5000'. Repeating the previous step to switch back to the default hierarchy results in the following table view:

	A	B	C	D	E	F	G	H
2	VERSION	Actuals *						
3	VERSION	Actual						
4	PROD	Unassigned	▼ SUV	Touran	Tiguan	Touareg	Van	Car
5	ACCOUNT							
6	▼ All	-	-5,000.00	-	-	-	-	-
7	Items So..	-	5,000.00	-	-	-	-	-
8	Stock	-	-	-	-	-	-	-

Image 6: Parent node with booked data

The parent node SUV now has data booked to it. None of the data has been assigned to its child members.

In the Modeler, when you update a model while a parent member in any hierarchy has data booked to it, a message appears:

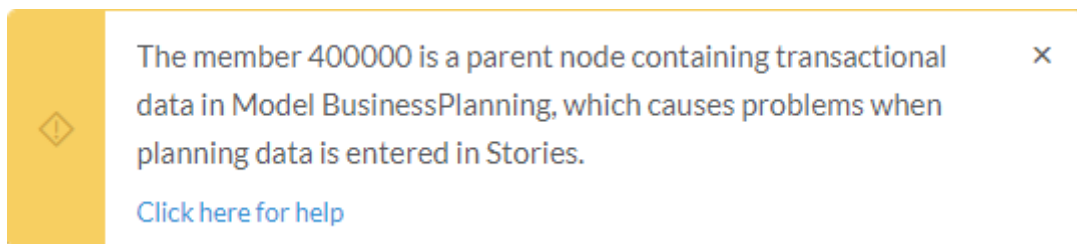


Image 7: Warning message during Model changes

RESOLVING THE PROBLEM

There are two ways to resolve the problem:

- Change the hierarchy. In this example, you would make SUV a leaf member in both hierarchies of the Prod dimension.
- Use the Distribute tool to move the data to leaf members of the dimension.

The second option allows you to resolve the problem without losing data. Follow the steps in the next section to implement this solution.

Distributing data to leaf members

Preparing a table

Open a table based on the model and identify the source and targets members for the Distribute operation:

- The source member is the parent member that has data booked to it, in this case, SUV.
- The target members must be leaf nodes in all of the dimension's hierarchies, and siblings of the source member in at least one hierarchy. In this example, the children of SUV in the default hierarchy will be used as source members. The Unassigned member can also be used as a target if you are not sure where to distribute the values.

The source and target members must be visible. To make the operation simpler, you can also filter the table to exclude other members:

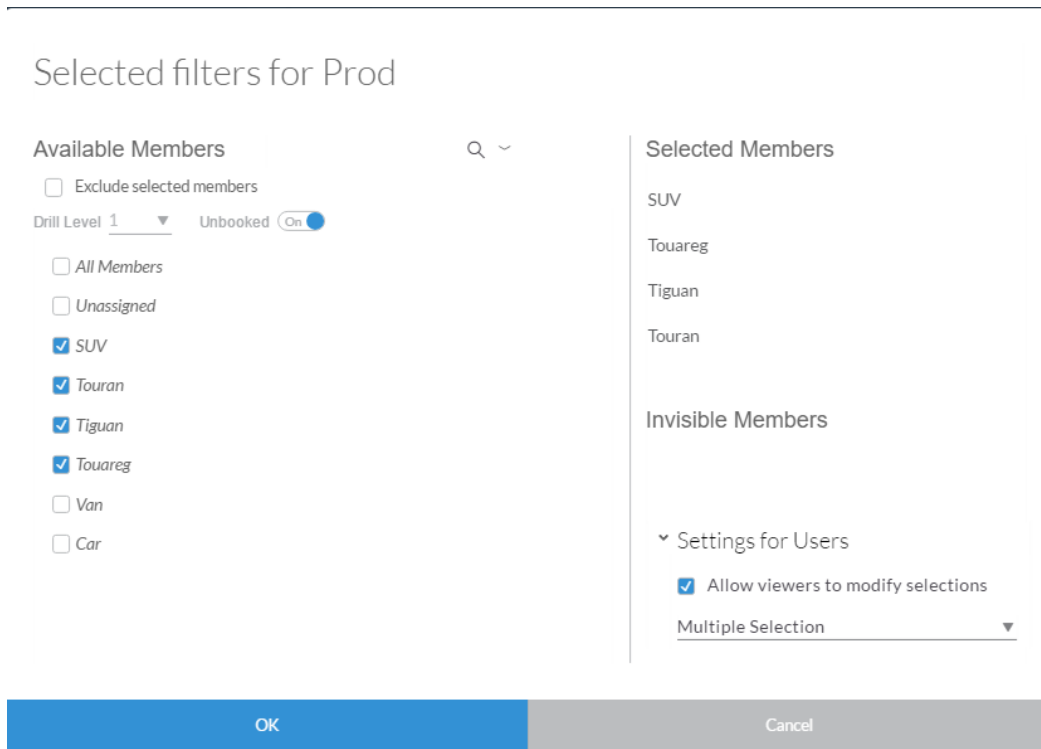


Image 8: Filtering the Prod dimension

Because the source and target members must be siblings, the flat hierarchy must also be displayed:

	A	B	C	D	E	F
1	DayModel in USD ☰ Prod Filter					
2	VERSION	Actuals*				Totals
3	VERSION	Actual				Total
4	PROD	SUV	Touran	Tiguan	Touareg	
5	ACCOUNT					
6	▼ All	-5,000.00	-	-	-	-5,000.00
7	Items So..	5,000.00	-	-	-	5,000.00
8	Stock	-	-	-	-	-

Image 9: The filtered table showing the flat Prod hierarchy

The table is now prepared for the Distribute operation.

Distributing the value

Select the source cell (Items Sold for SUV, cell B7) and select  > **Distribute**.

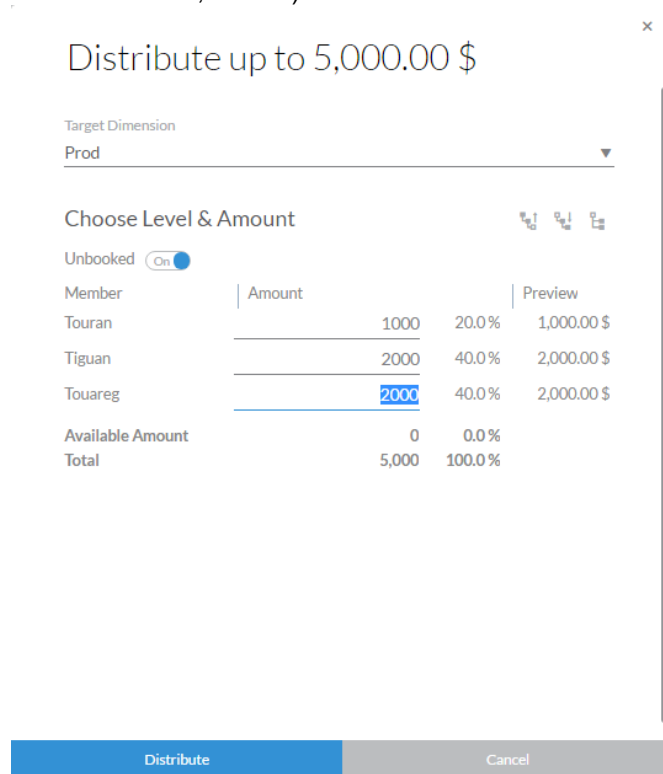


Image 10: Distribute dialog

The Distribute dialog allows you to move part or all of the value to siblings of the source member. Note that it shows absolute values rather than weights.

Distribute the entire source value among the target members by typing values in the Amount field, and then select **Distribute**.

	A	B	C	D	E
2	VERSION	Actuals *			
3	VERSION	Actual			
4	PROD	SUV	Touran	Tiguan	Touareg
5	ACCOUNT				
6	▼ All	0.00	-1,000.00	-2,000.00	-2,000.00
7	Items So..	0.00	1,000.00	2,000.00	2,000.00
8	Stock	-	-	-	-

Image 11: Distributed values

The value of the source cell should now show 0.00 in the flat hierarchy. In the default hierarchy, the leaf members should aggregate up to the parent node as usual.

Changing a model to prevent further problems

The hierarchies can be adjusted in the Modeler to prevent the problem from occurring again. You can avoid the error completely by following this rule:

No parent member of any hierarchy can be a leaf member in any other hierarchy.

If it is necessary to work with a hierarchy where all parent nodes are removed, you can add a placeholder member to the dimension and make all the non-leaf nodes children of this member.

These non-leaf members will still be visible in filter dialogs and when unbooked data is visible, so it is recommended to use a clear description for the member to prevent users from booking data to its children, such as “Do not enter data here”.

For example, the ParentNodes member has been added to the Prod dimension in these images, and the non-leaf nodes in the default hierarchy have been grouped together as its child members to make unintentional data entry less likely:

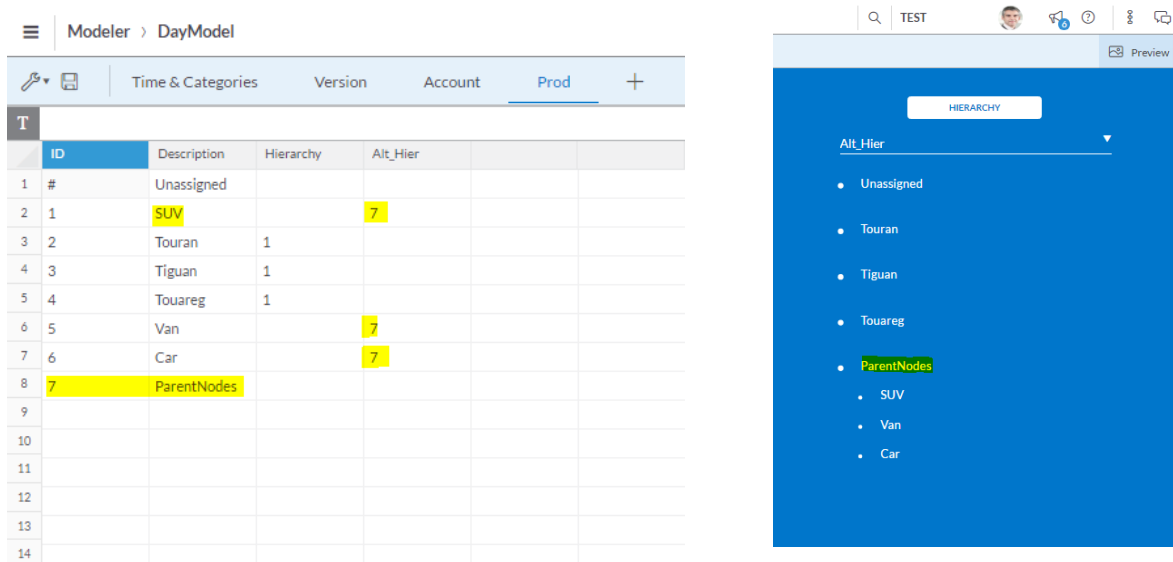


Image 12: Model with non-leaf nodes separated