SAP BW IP/BPC

BW - IP Building blocks

Capgemini

Course Objective

What to expect from this training:

- Understand the basic planning concepts
- Position SAP BW Integrated Planning as a enterprise solution for business planning

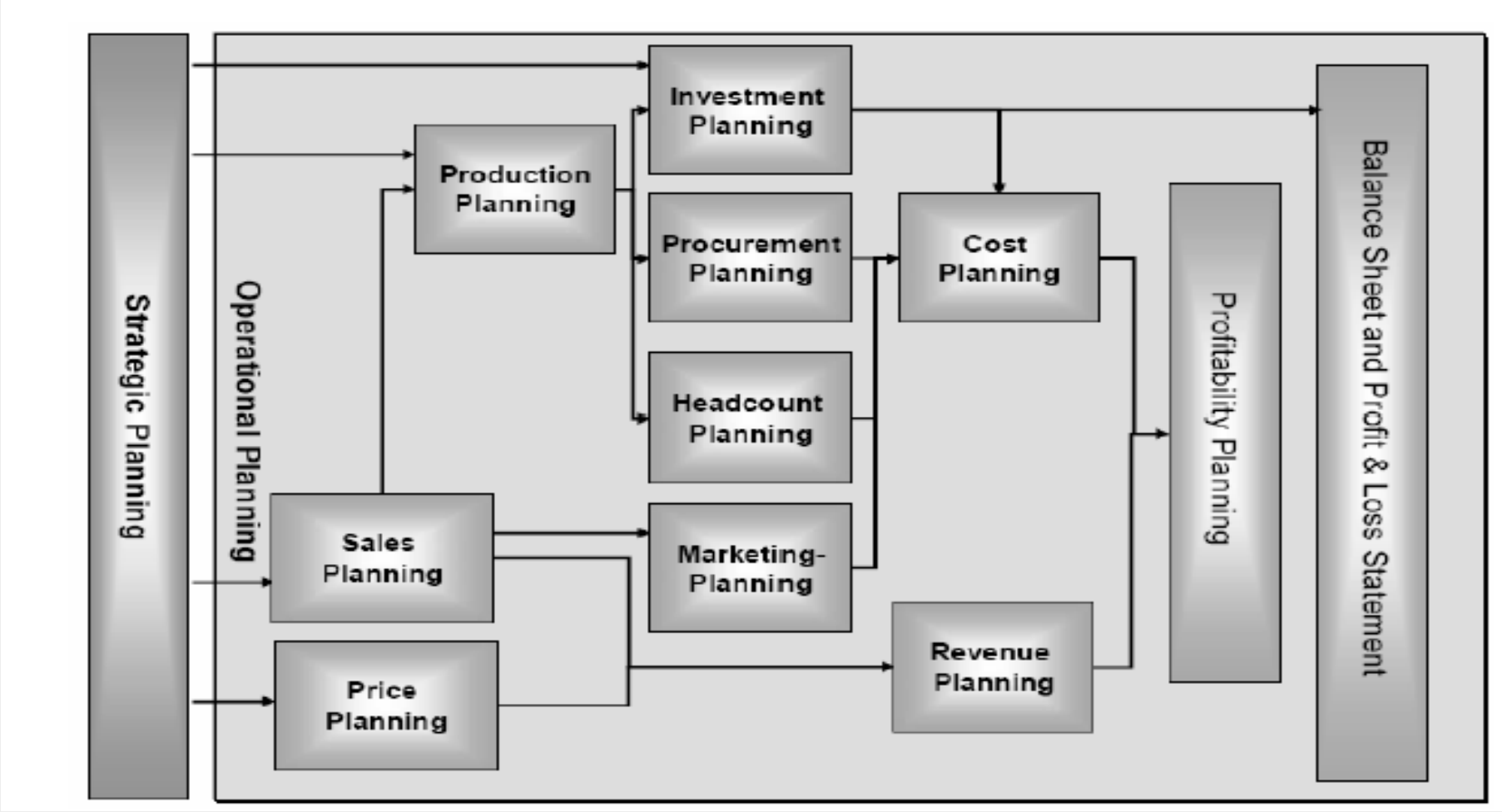
What is SAP BW-IP ?

With SAP BW-IP, SAP delivers tools for generic planning, supporting a planning process that allows for the construction of a planning model. This model integrates global strategic planning and specific operational planning problems in a closed-loop process, with a strong focus on business processes across the enterprise.

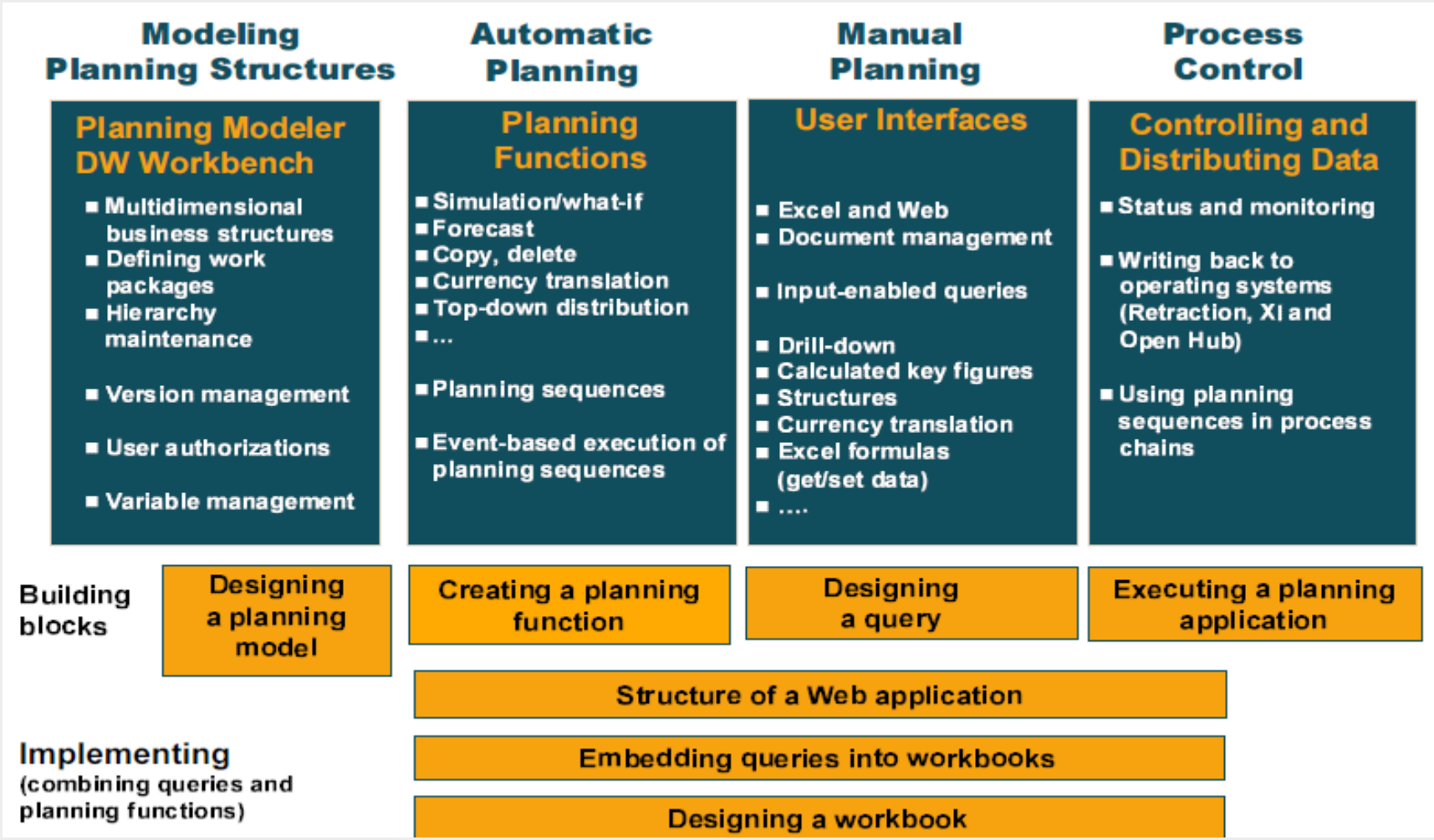
SAP BW-IP helps you plan, budget, and forecast by providing functions such as:

- Top-down planning and bottom-up contribution with a rich set of planning functions.
- A planning framework that lets you create and maintain planning models.
- A user interface for manual planning and analysis.
- Tools for process control (i.e., status tracking and monitoring).

Types of Plan



Integrated Planning building blocks



Integrated Planning – Modeling

- **Real time InfoProviders**
- Aggregation Levels
- Filters
- Planning Functions
- Planning Sequences
- Characteristics Relationship
- Data Slices
- Input Ready Queries

InfoProviders – Integrated Planning

Real-Time InfoCubes:

- Created using the Data Warehousing Workbench
- DataStore object for plan values
- Basis for planning
- Data load Behavior: data load can be loaded
- Can be part of MultiProvider

A real time InfoCube is a special Basic InfoCube, especially developed for planning applications.

The system accesses data in such a InfoCube is transactional, in other words, data is written to the InfoCube and instantly read again when required.

- e.g. In BW BPS/IP, users enter plan data via planning layouts which is in turn stored in Real time InfoCubes.

A standard Basic InfoCube is optimized for pure read access and hence not suitable here.

Infoproviders used in Integrated planning : Real time Cube

BW integrated planning can use only real time InfoProviders for planning purpose (that is, changing or writing data); these InfoProviders can be assigned to a MultiProvider.

The real-time InfoProvider can be filled here either manually or by planning functions.

Data requests in real-time InfoProviders stay open until the number of records in the data request exceeds 50,000. When an application writes data to the request and the limit of 50,000 records is exceeded, the data is stored and the request is closed. Thus, a request can actually contain more than 50,000 records.

A real-time InfoProvider can be either loaded or planned, meaning that it has two " states" .

- **Real-time InfoCube can be loaded with information; no plans allowed**
- **Real-time InfoCube can be planned; no data loading allowed.**

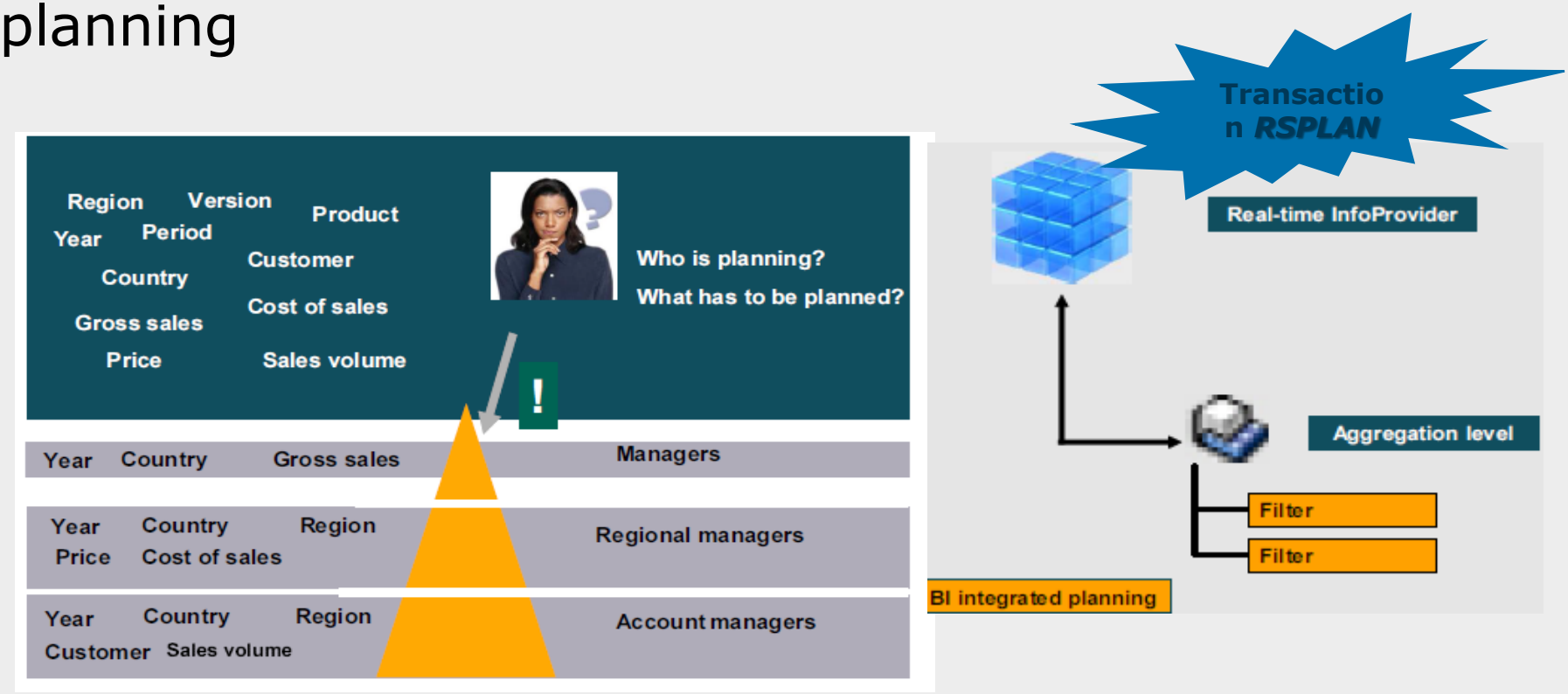
In the first case, the information can be loaded whilst the plan data can only be read and not changed. In the second case (planning), the InfoProvider cannot be loaded at the same time, nor can it be filled via a data transfer process.

It is not possible to transfer data to a second InfoProvider until the first real-time InfoProviders request has ended.

Integrated Planning – Modeling

- Real time InfoProviders
- **Aggregation Levels**
- Filters
- Planning Functions
- Planning Sequences
- Characteristics Relationship
- Data Slices
- Input Ready Queries

Aggregation level : Deciding the granularity of planning

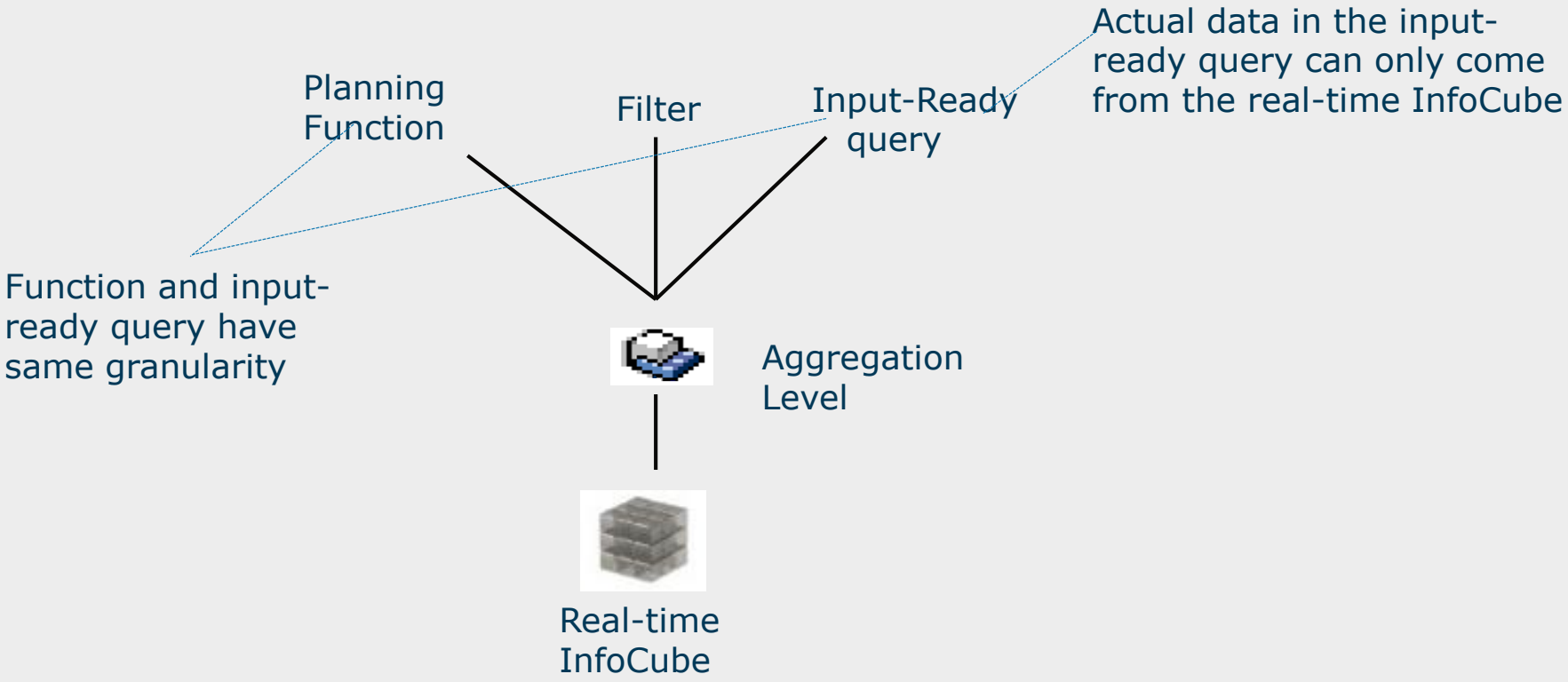


Aggregation levels are used as InfoProviders for planning: They allow you to model planning levels that can be changed manually, using input-ready queries, or automatically, using planning functions.

An aggregation level is defined by a set of characteristics and key figures of the underlying InfoProvider. The key figures contained in the aggregation level are aggregated using the characteristics not contained in the aggregation level

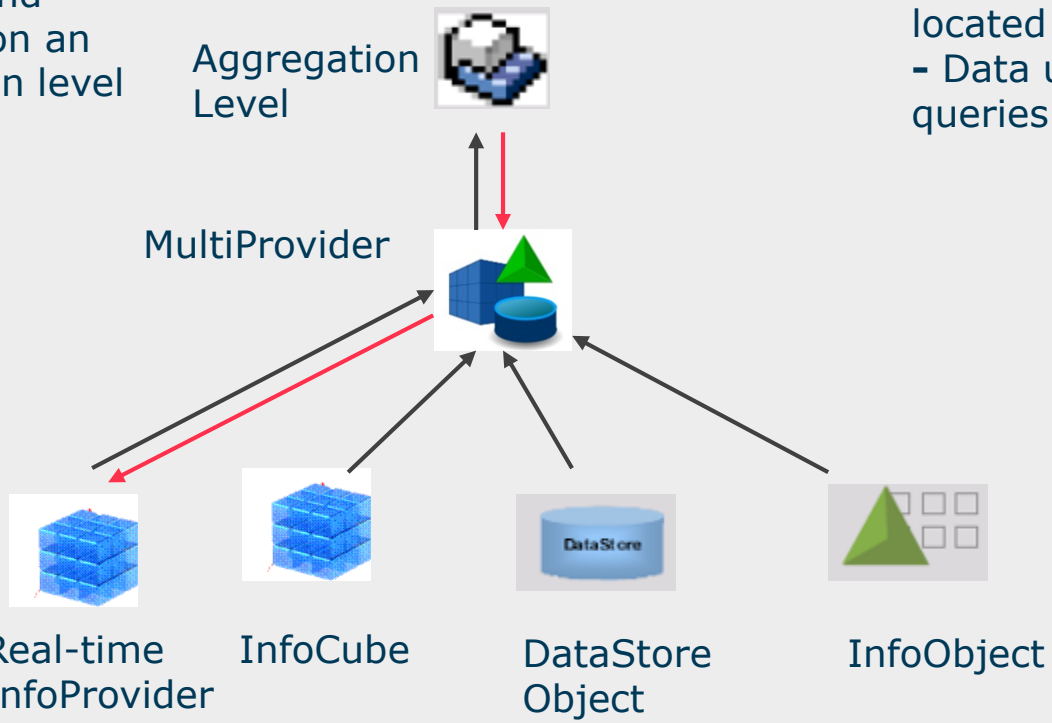
Aggregation level : deciding the granularity of planning…Continued

It is also possible to use rea—time InfoCubes both for loading actual data via extraction and for planning. To do this, you have to set a switch in the InfoCube to enable either the “loading” or the “planning”.



Aggregation level and MultiProvider: Modeling

Planning and reporting on an aggregation level



- Planning functions and queries are located on an aggregation level.
- Data used by planning functions and queries has the same granularity.

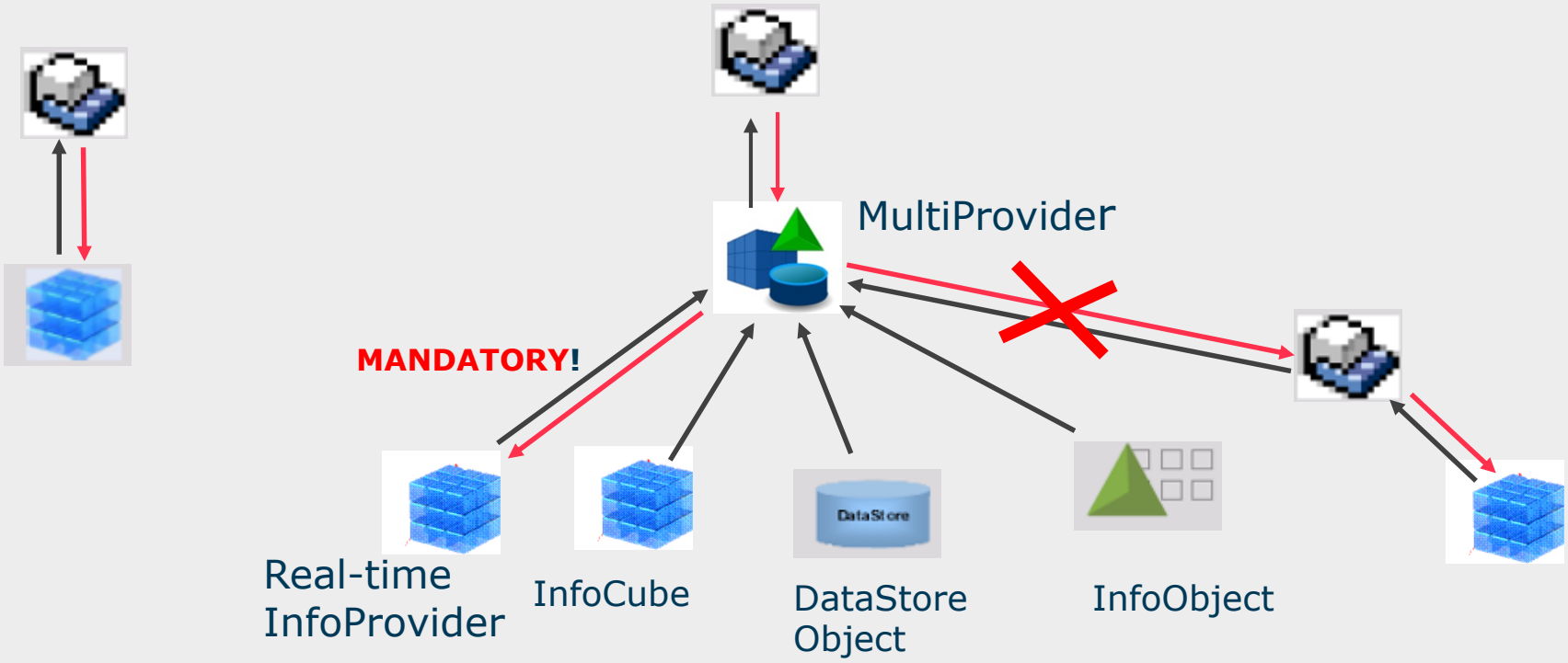
As you can see, all the aggregation levels are created on the MultiProvider, so that both the planning that is, the planning functions – and the plan queries can be set up on the same aggregation level. This reduces the frequency of errors and **the granularity of the planned and reported data is same!**

If reference data is required for the actual status, it can be read from the "InfoCube" or the "DataStoreObject",for example.

Aggregation : Modeling pre-requisite for Input ready query

Simple Aggregation Level

Complex Aggregation Level



Integrated Planning – Modeling

- Real time InfoProviders
- Aggregation Levels
- **Filters**
- Planning Functions
- Planning Sequences
- Characteristics Relationship
- Data Slices
- Input Ready Queries

Filters

A filter is an object that describes a multidimensional segment of data from a dataset.
Filters are used in reporting, analysis and planning to restrict data to a certain business area, certain product group, or certain time periods, for example.
Segmenting the dataset in this way can ensure that users or user groups only have access to the data that is relevant to them, or that only certain data areas are visible within an application scenario. Within BI
Integrated Planning, filters set the selection for the data upon which a planning function operates.

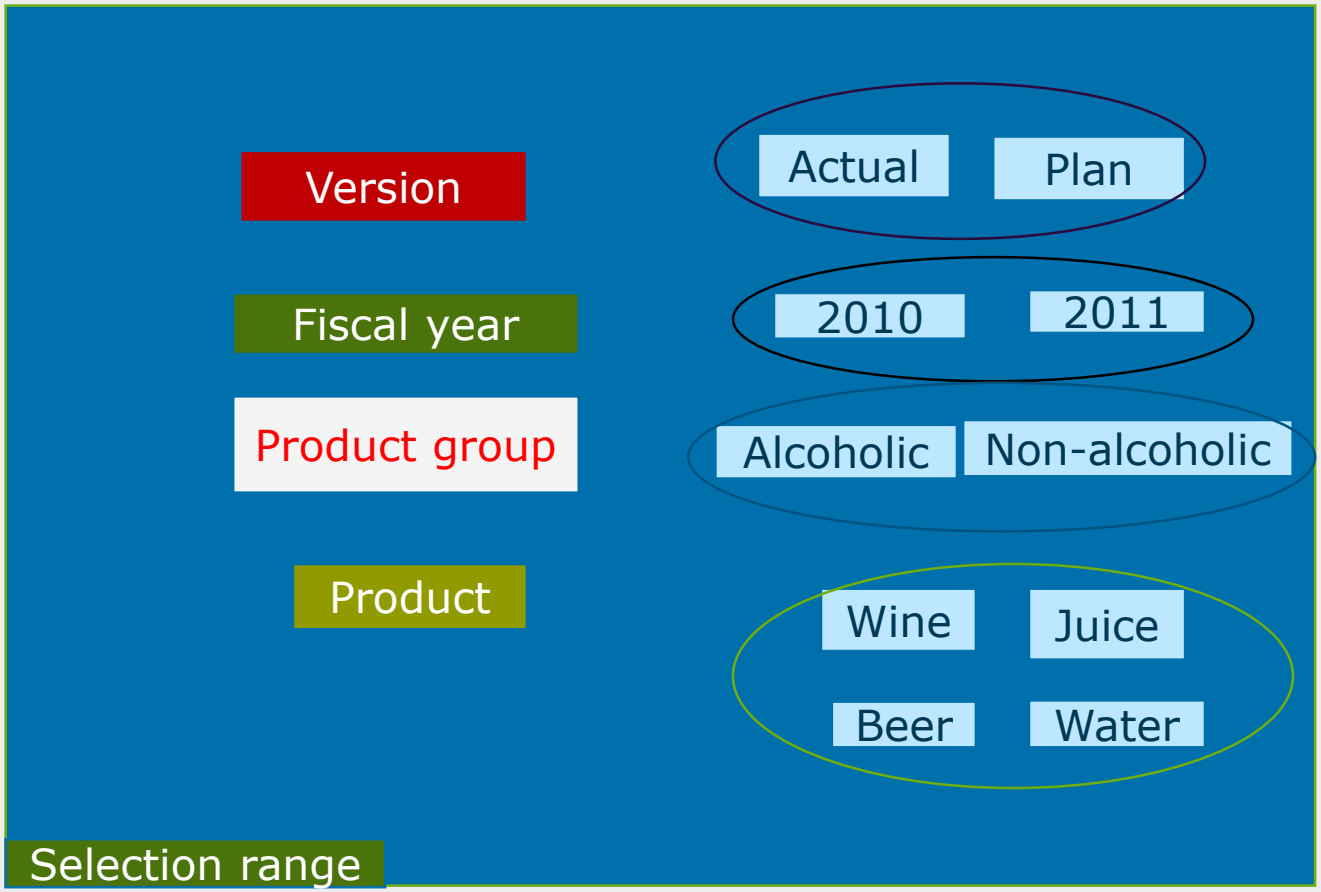
Filters are mandatory if you want to use a planning function in a planning sequence. They are optional if you only want to create input-ready queries using the aggregation level.

You can create multiple filters for an InfoProvider. To do this, you can use the Planning Modeler or the Planning Wizard as well as the Query Designer.
You can use the "**Show Advanced Settings**" button to limit selected characteristics of the InfoProvider to single values, value ranges, hierarchy nocks, history or favorites and you can specify whether or not these can be changed when the query is executed.

You can enter a filler key date for specifying selections, such as determining a time-hierarchy of tine-dependent hierarchy node selections. You can use the standard provided variable OPLANDAT on characteristic OCALDAY to synchronize key dates in queries, filters, characteristic relationships, data slices and planning functions. That way, you can ensure that the same key date is used in all the objects.

Filters as selection range

The filter specifies which data a user is allowed to plan with



Functionality of a filter:

A functionality depends on its particular use in either a planning function or a query.

Filters in Planning Functions

In connection with planning functions, a filter in the characteristic restrictions describes the data that is used for executing a planning function. Selections in the default values are **not used for executing the planning function**.

You can also use a key date for tip filter to determine time-dependent selections.

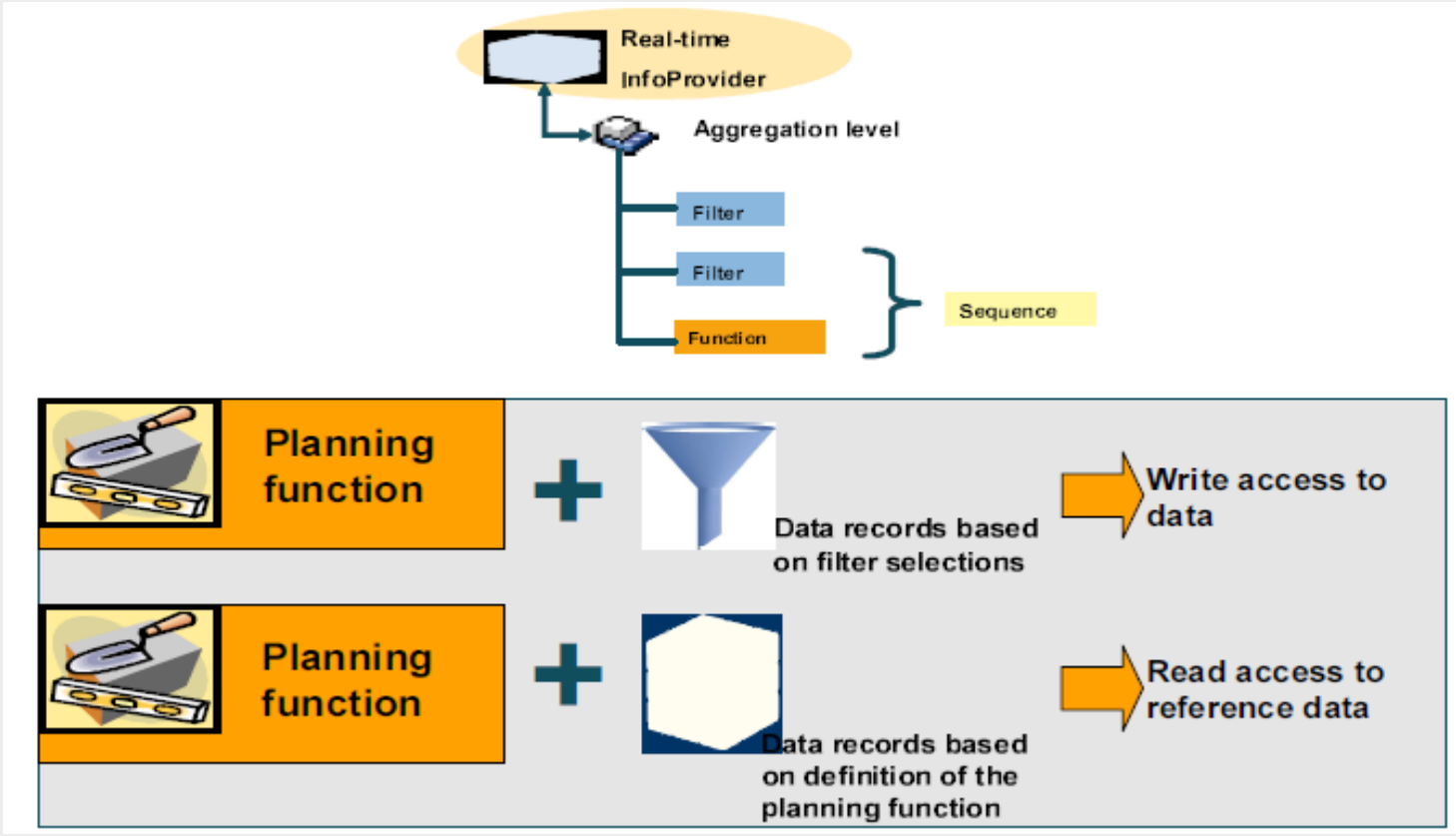
Filters in a Query

The values defined in a query's characteristic restrictions limit the volume of data that is available for further filtering during a query runtime. It is then no longer possible to filter using a characteristic value outside this value set. TIP default values determine the query’s initial filter state.
The settings Changeable on Execution and Single Value Only essentially refer to the use of filters with a query:

• Changeable on Execution determines whether the selection of values affected by the characteristic restrictions can be changed when the query is executed. This setting is a prerequisite for tip definition of &fault values for a characteristic.

• If you select the Changeable on Execution option, you can use the Single Values Only option to specify that only a single value can be used for filtering the query.

Filters in planning functions and importing reference data:



The filter specifies the write access's selection range for planning functions. Selections of data records that are contained in the InfoProvider but not stored in the filter can always be imported as reference data, however it is not possible to write access them.

Variables

Object	Variable used for:
Queries, particularly input-ready queries	-Parameterization of characteristic restrictions in the Query -in formulas, conditions, exceptions, placeholder for texts
Filter	Parameterization of characteristic restrictions
Planning functions	Depending on the planning function type for parameterizing conditions and parameters
Characteristic relationships	-For parameterization of the hierarchy -For parameterization of the selection from a datastore object
Data slice	Parameterization of characteristic restrictions that describe the data slice
Additional objects	Parameterization of a display hierarchy for the query

Variables…..continued

Variables are used to parameterize a query, a planning function, a filter, a characteristic relationship or a data slice. When you execute a query, planning function or Web application, these placeholders are filled with values.

Variables act as placeholders for characteristic values, hierarchies, hierarchy nodes, texts, and formula elements, and can be processed in many different ways.

• There are different variable types, depending on which objects you want to define variables for.

• The processing type specifies how a variable is filled during a query's, a planning function's or a Web application's runtime.

Variables do not depend on an InfoProvider. they only depend on the InfoObject in question. A variable that was defined for an InfoObject is available in all the InfoProviders that use this InfoObject.

You can define variables in the Query Designer or the planning modeler or Planning Wizard. The variables are then available for re-use in all queries or planning functions.

Please note that you can only ever assign one characteristic to a variable. Also note that there are no user-dependent variables in BW integrated planning. If you require user-dependent variables, you should create a DataStore object containing the user master data and then have the DataStore object upload the user-specific characteristic combinations with a flat file. Then you have to create a query on the DataStore object that can be used to access the user-dependent combinations.

Variables…..continued

There are different variable types, depending on which objects you want to define variables for.

Characteristic value variables represent characteristic values.

Hierarchy variables variables represent hierarchies.

Hierarchy node variables represent a node in a hierarchy.

Text variables represent a text. You can use text variables in query descriptions, calculated key figures and structural components.

You can use text variables when creating calculated key figures, restricted key figures, selections and formulas, in the description of these objects.

Formula variables represent numerical values. You can use formula variables in formulas. In addition to this, formula variables are used to define exceptions and conditions.

Variables…..continued

Manual Entry/ Default Value
Replacement Path
Customer Exit
SAP Exit
Authorization

Manual Entry: allows you to enter the necessary value manually or to use the default value specified when the variable was defined.

Replacement Path : You can use this to specify the value that automatically replaces the variable when you execute the query or Web application.

Customer exit: is a predefined enhancement that you can develop using customer-specific logic.
The prerequisite here is that you created a project in transaction CMOD and that you have selected SAP enhancement RSR00001 for global variables in reporting and you have assigned it to the enhancement project. You then activated the project.

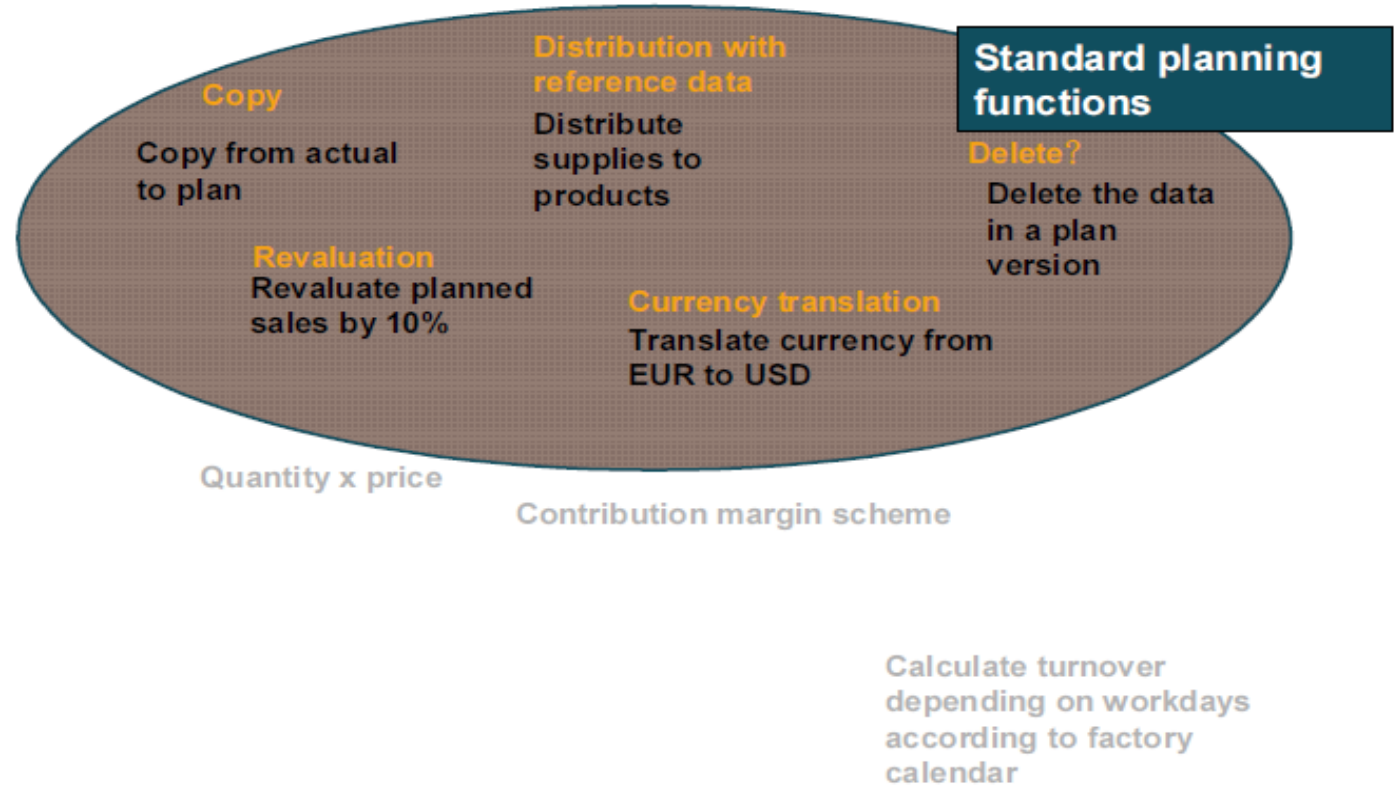
The SAP Exit processing type is contained in variables that are delivered as part of the Business
Content.

The **Authorization** processing type enables you to fill the variables automatically with the values from
the user's authorization

Integrated Planning – Modeling

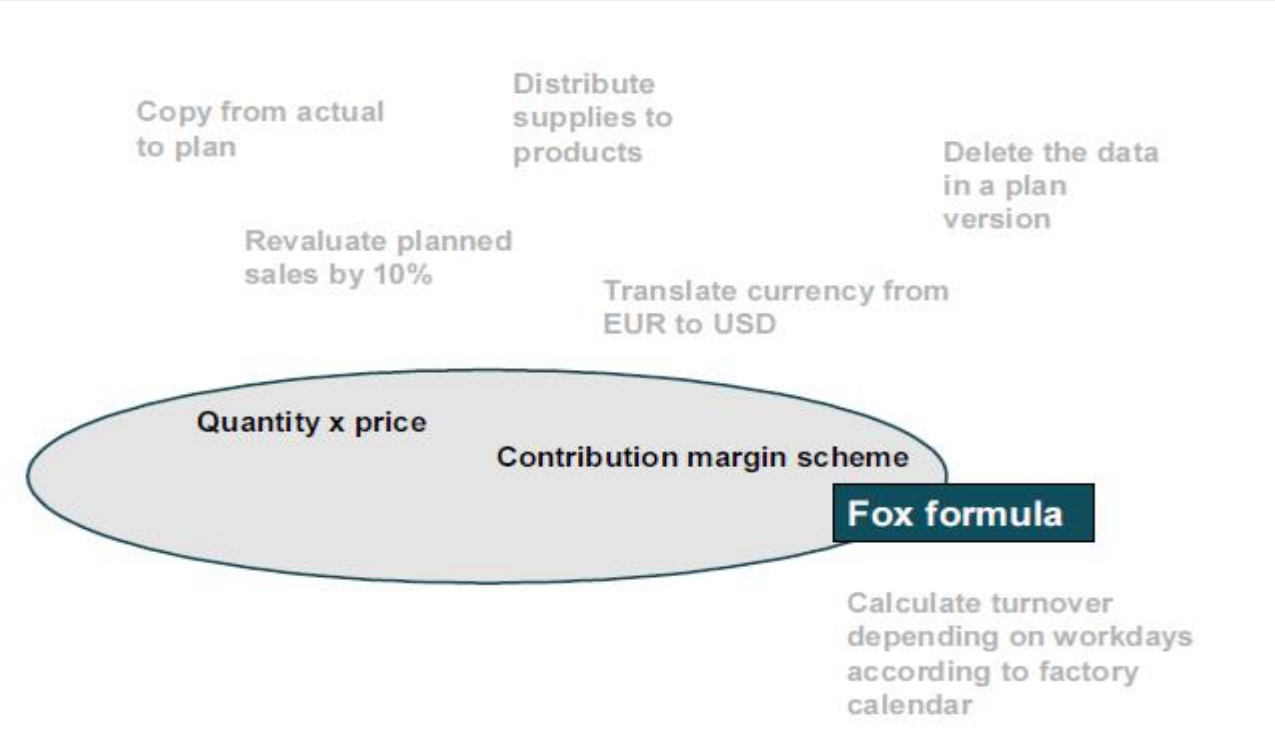
- Real time InfoProviders
- Aggregation Levels
- Filters
- **Planning Functions**
- Planning Sequences
- Characteristics Relationship
- Data Slices
- Input Ready Queries

Planning Functions: Various business requirements - Scenario 1



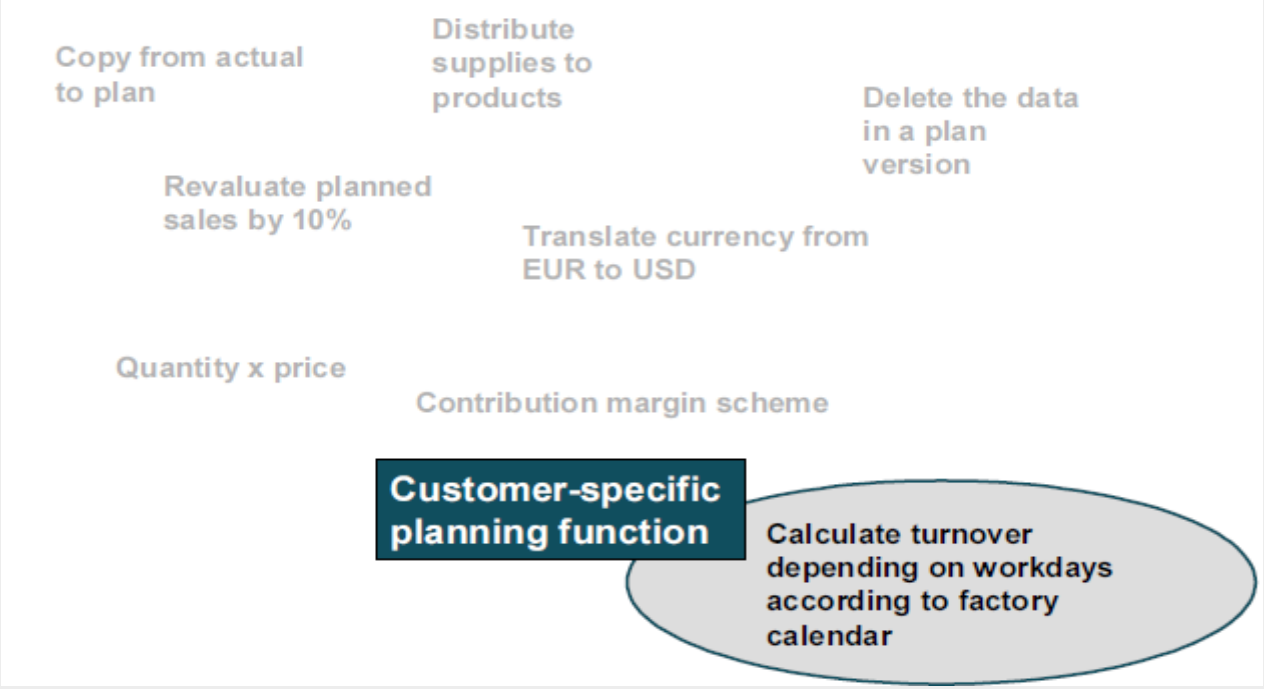
In a typical scenario, there is a need to for a wide range of automatic functions to change or create plans. BW Planning offers different types of planning functions that can be used to create all necessary automatic functions: Standard planning functions with fixed behavior that can be easily setup.

Planning Functions: Various business requirements - Scenario 2



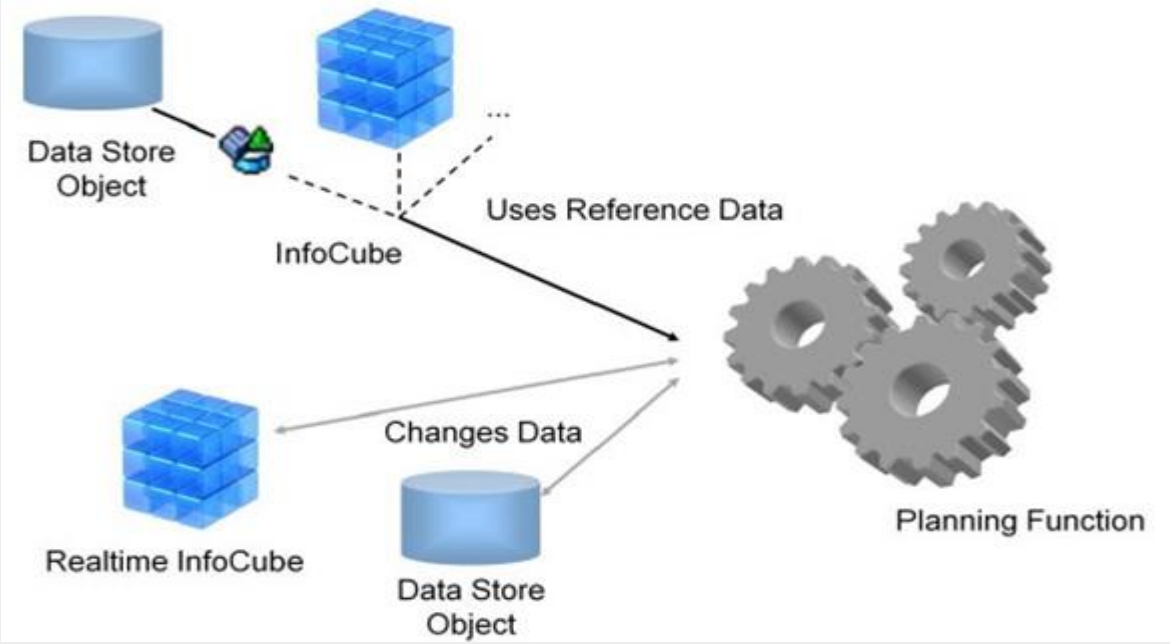
Whenever a planning function cannot be achieved using a pre-defined planning function type, you can use Fox Formulas to create your own planning functions. With Fox Formulas, BI Planning provides a toolkit for creating simple
calculations (such as a volume times price calculation) as well as a simple to use yet very flexible and powerful tool for complex formulas. No programming (ABAP, Java, etc.) is necessary when using Fox Formulas. Above you can see some examples for planning functions that can be created using Fox Formulas

Planning Functions: Various business requirements - Scenario 3



When a planning function cannot be achieved using one of the predefined types or a Fox Formula, you can create your own type of planning function (using ABAP). These types of planning functions can have their own screen in Customizing in the planning modeler and can be reused throughout the system.

Data Selection in Planning Functions: Reference & Transaction data



A planning function can (potentially) use two types of plan data: The data to be changed by the planning function and the data that the planning function uses as additional information upon execution. We call the first set of data ‘transaction data‘ and the latter ‘reference data‘.
Transaction data: A planning function can only change data that is stored in a real-time InfoCube. The logical level of aggregation of the data records is defined by the aggregation level (in the planning modeler) upon which the planning function is defined. The transaction data is locked upon execution of the planning function..

Planning Function - Copy

Copying from version 1 to version2

Initial data record

New data record

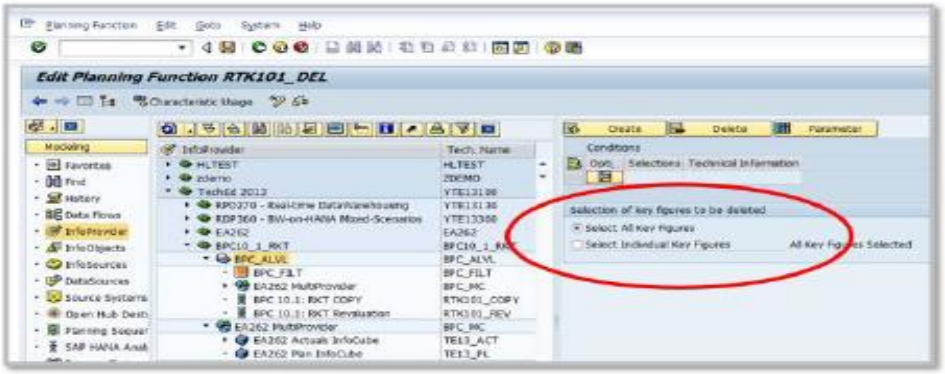
Division	Fiscal year	Version	Amount	Quantity
C-DIV-61	2010	1	100	10
C-DIV-61	2010	2	100	10

"Characteristic to be changed"

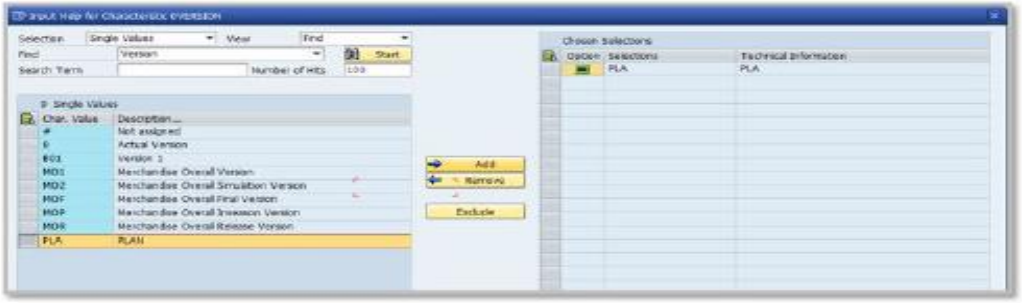
The copy function is used to copy actual data from an InfoCube to a real-time InfoProvider.
The parameter records contain the actual rules for copying. These are specified in the " From Field" and the "To field" Note that the copy function is always set to overwrite the target data records if a planning function is executed more than once. It is possible to copy from a source characteristic to a target characteristic, to a target interval, to a hierarchy node or to a variable.
If you saved multiple simultaneous attributes for a sender characteristic and you copy to a characteristic attribute of a target characteristic, all the sender data records are summarized and copied to the target attribute.
You can decide whether all the key figures are to be copied or only individual ones.

Planning Function - Delete

When creating a delete function, only key figures are changed. This means once again that we do not need to specify any characteristics that are to be changed. As with the other planning functions, you can specify whether you want to delete all key figures or only individual ones. Make sure that you run the delete function at the proper level of aggregation. If you want to make sure that all records in a given selection are deleted, you have to choose an aggregation level that contains all characteristics.

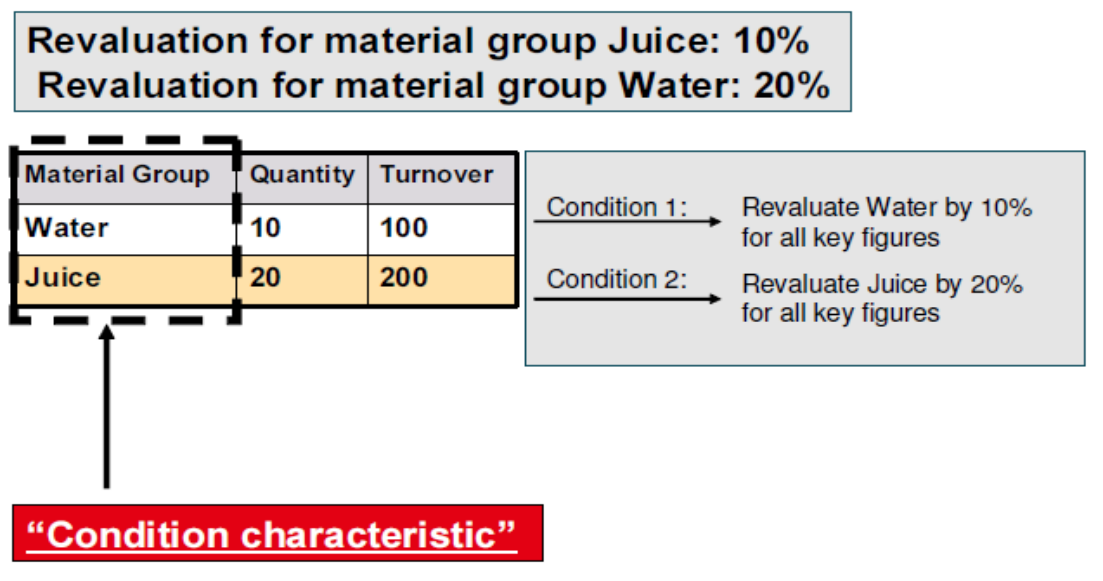


Selection of key figures to be deleted



Select version PLAN1

Planning Function - Revaluation

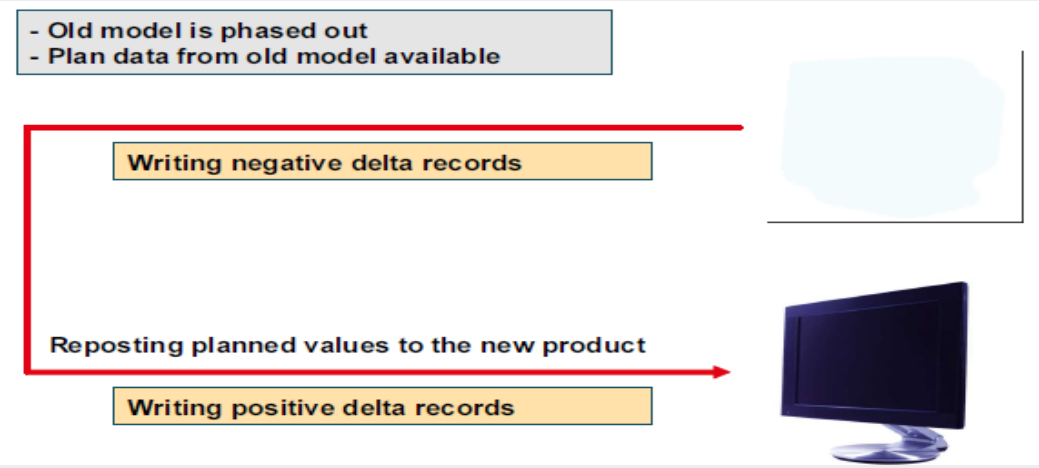


Condition characteristics are the characteristics that are not included in the " characteristics to be changed" . For these, you can define a condition, although you do not have to.
In the screenshot above with the revaluation function, different conditions are to be defined for the individual material groups. To make this possible, the Material Group field has to be included in the " condition characteristics" .
Only characteristic conditions can be specified for defining the revaluation function.

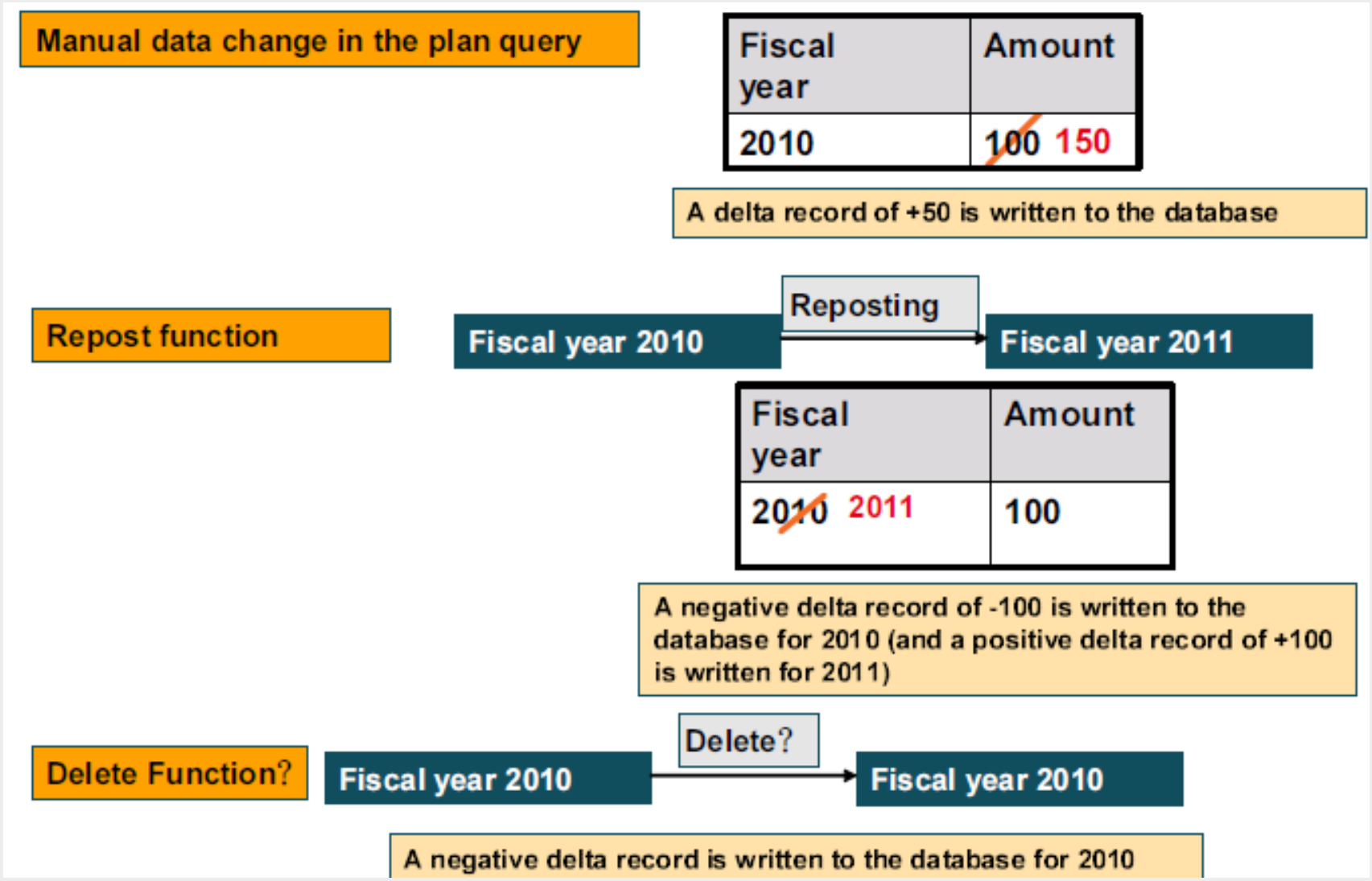
Planning Function – Repost

You can use the Repost function type to post the key figure values from existing characteristic combinations to other combinations (similar to the copy function). Unlike copying, the repost function writes credit lines that are offset against the original debit to zero in the buffer.

Typically, you would use this function to remove an account assignment error.
You can use the key figure selection table to determine which key figures are reposted.
You can use the Before and after value for the reposting procedure table to create either a simple reposting procedure or various reposting procedures in a planning function.
The following rules apply:
· Both the before and after values for the reposting procedures have to be singe values.
· Both the before and after values are changed and therefore have to be in the filter that is transferred to the planning function.
· During reposting, the key figure values are always added to the after values.

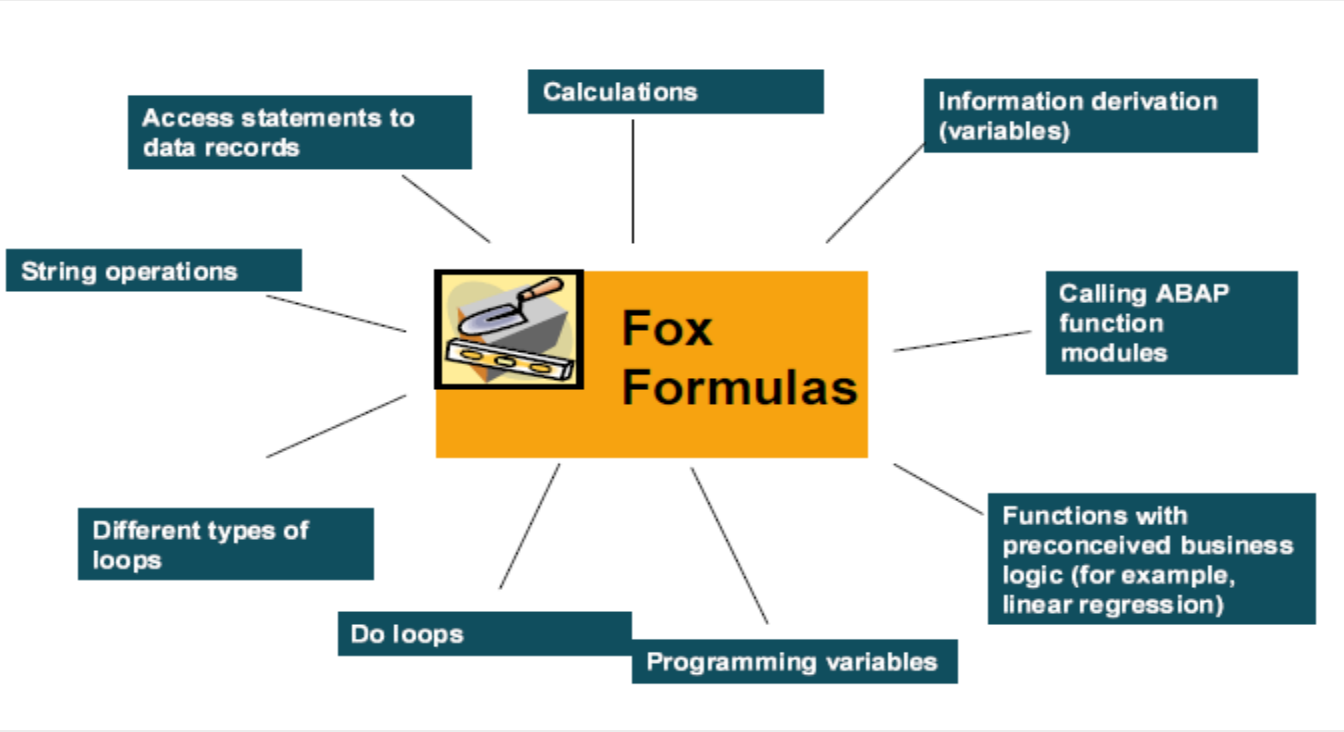


How are data records changed ?

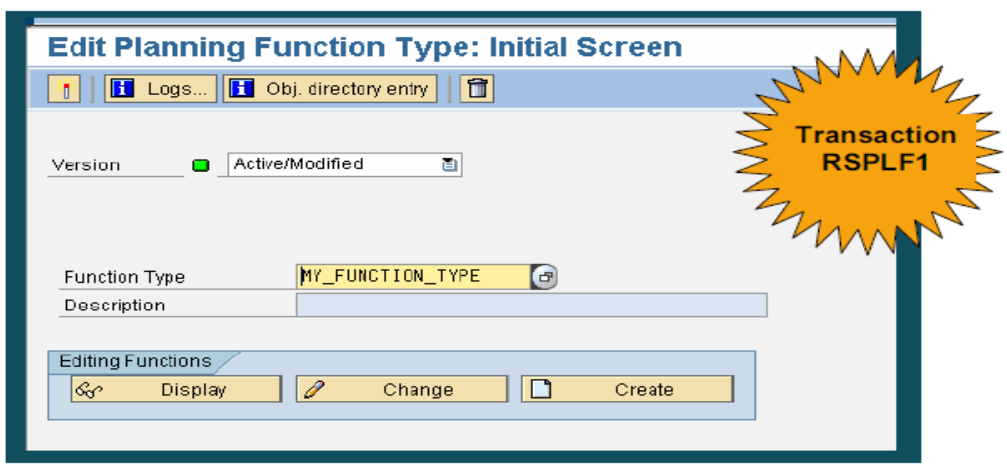


Formula Extension (FOX)

With the function type FORMULA (FOX = Formula Extension), you can use enhanced mathematical functions for calculating the plan data. ABAP knowledge is not required. From the FOX coding the system generates ABAP coding that is executed at runtime



Custom Planning function types



Create a custom planning function type if…

- The logic in FOX would be very complicated
- You need several complex FOX formulas for doing the job
- You need syntax elements that are not contained in FOX
- You need the same planning function in different levels. You can use the same function type in several places instead of copying a predefined or formula function to the different aggregation levels.

Custom Planning function types : A typical use

- In a retail company, sales increase on Saturdays and on the day before a holiday. A distribution function is therefore required to distribute the sales over the calendar month on the basis of the number of Saturdays and holidays in the month.

- In the central IT department, certain planning functions are set up as content so that these functions can be used in different projects.

Every planning function type consists of a definition part (meta data), which is created and changed in a transaction (RSPLF1), and an ABAP OO class, in which the actual procedure is programmed. The class name is part of the definition part.

Please note that all the planning functions supplied by SAP are defined in the same way, like the customer-specific planning function types, so that the former can be used as examples for creating customer-specific functions.

The function type Delete (0RSPL_DELETE) is a simple example. There is only one parameter (KYFSEL) for selecting the key figures to be deleted. In the related ABAP class, the two interfaces IF_RSPLFA_SRVTYPE_IMP_CHECK and IF_RSPLFA_SRVTYPE_IMP_EXEC are implemented.

Integrated Planning – Modeling

- Real time InfoProviders
- Aggregation Levels
- Filters
- Planning Functions
- **Planning Sequences**
- Characteristics Relationship
- Data Slices
- Input Ready Queries

Planning Sequence:
Combining planning functions and filters

- Planning sequences combi1P planning functions and filters, whereby a sequence Can comprise one or more combinations of planning function and filter.

- A planning sequence is mainly defined for three reasons:
  • To bundle multiple planning functions into one complex function and automatically execute them in one step
  • To test one or more planning functions at runtime
  • To execute a planning function or a sequence as a batch process within a process chain

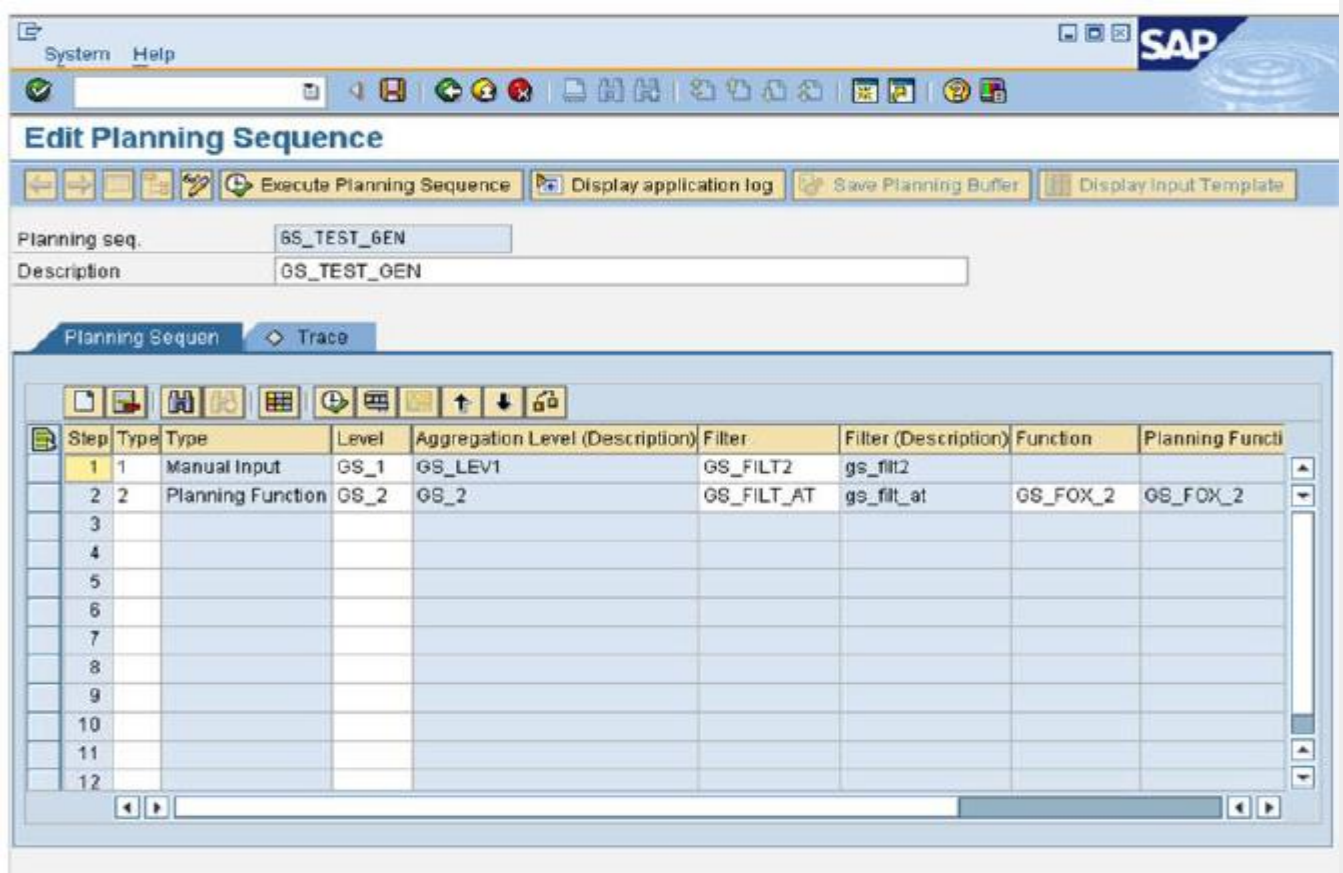
- Within a sequence, a combination of planning function/sequence is known as a "step".

- For each step, you always have to enter the underlying aggregation level, the filter, and the planning function.

- You can use a planning sequence to bundle planning functions from different aggregation levels.

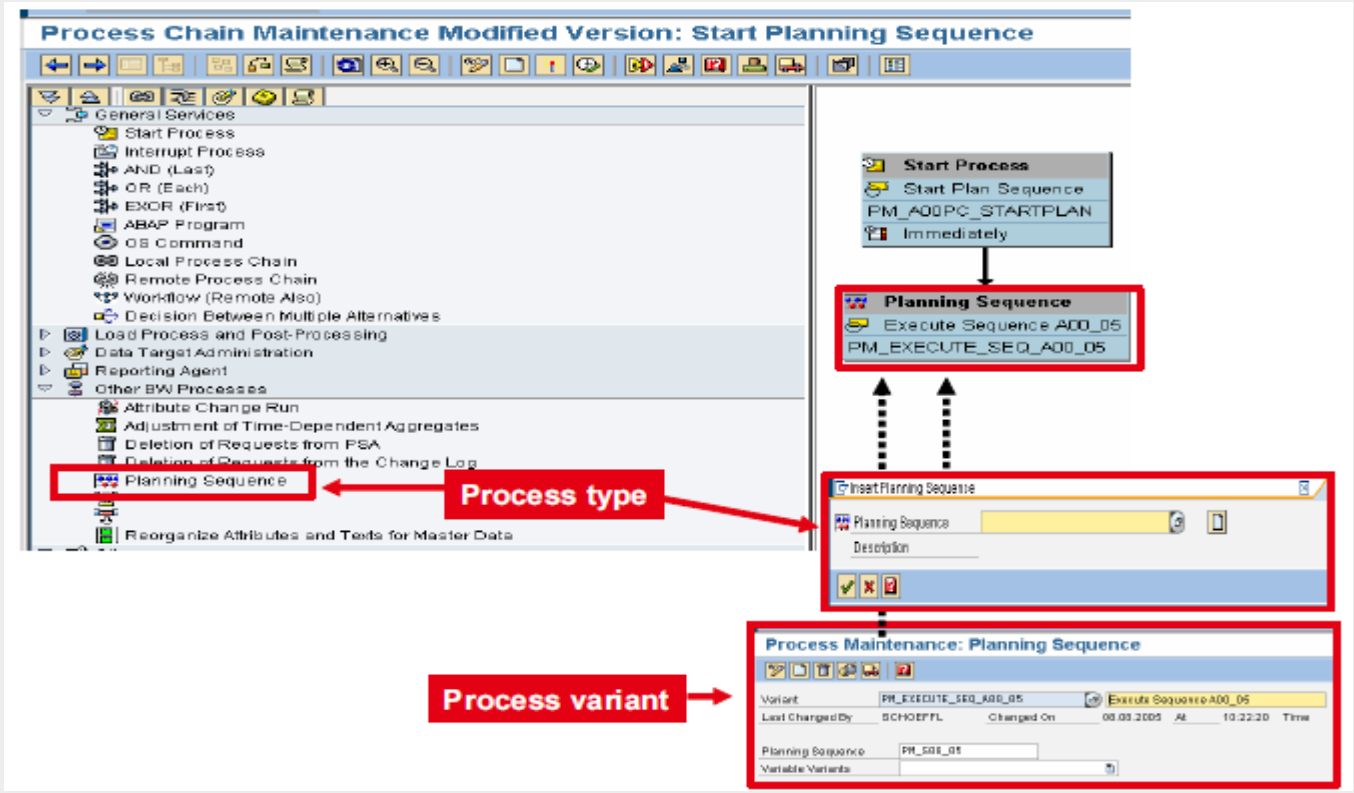
- It is not possible to include one planning sequence in another.

Planning Sequence: Combining planning functions and filters



Planning Sequence: Batch execution of planning sequence

If you want to execute different sequences in a process chain, you must create a step for each planning sequence, and the steps are then combined. Every step has its own data buffer, and at the end of each step the data is automatically saved, after which the plan buffer is released and the data is unlocked



Integrated Planning – Modeling

- Real time InfoProviders
- Aggregation Levels
- Filters
- Planning Functions
- Planning Sequences
- **Characteristics Relationship**
- Data Slices
- Input Ready Queries

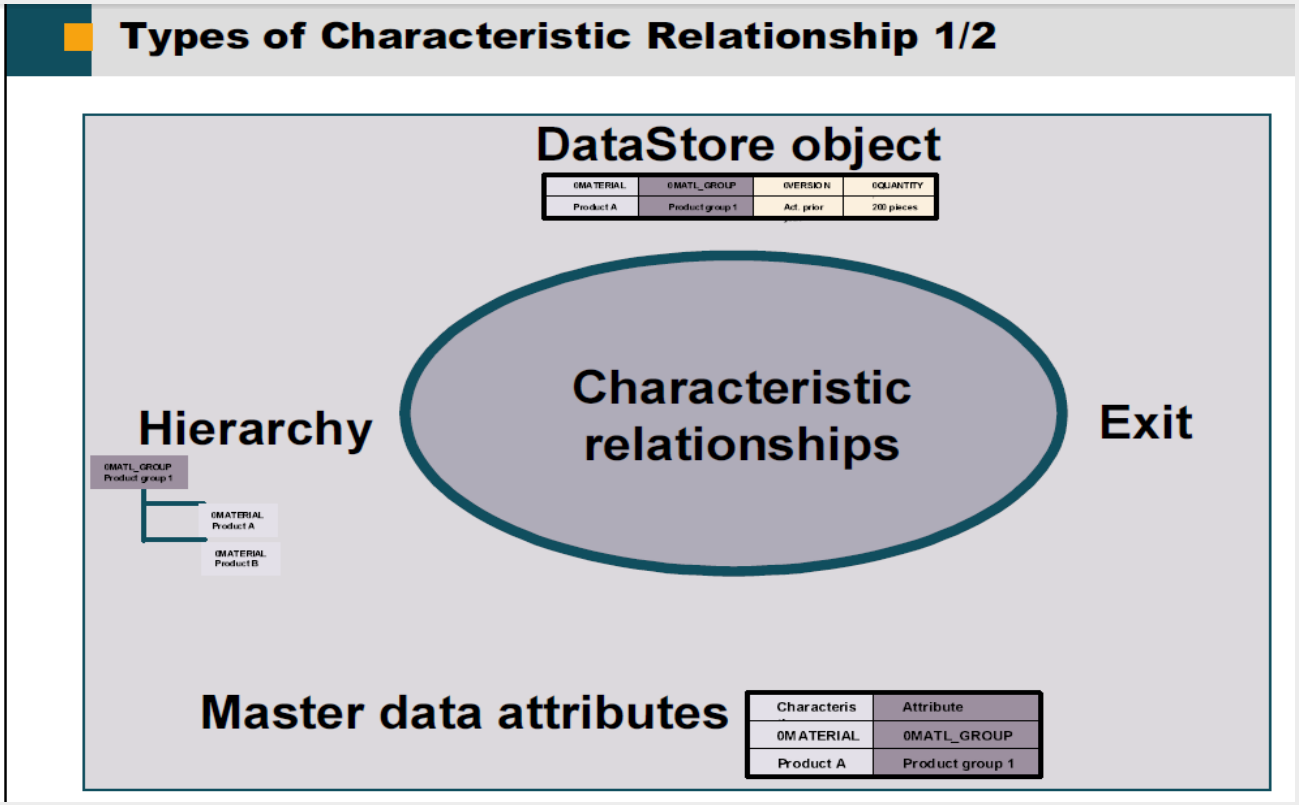
Characteristic Relationship

Characteristic relationships:

- Are the basis of the data model
- Apply per real-time InfoProvider and specify rules between characteristic attributes
- Ensure data consistency regarding combinations of characteristics

If you are working with MultiProviders, they are subject to the characteristics that were set up for the lower-level real-time InfoProviders.

Characteristic Relationship



Characteristic Relationship

➢It is used to test valid combination of records generated and derive the char values for other chars. This is at info provider level; it is applicable to all the aggregation levels of real time cube.

➢Invalid combination: this is applicable in manual planning, when we are trying to generate a new combination which is not part of Char relation, it is assumed as invalid combination.

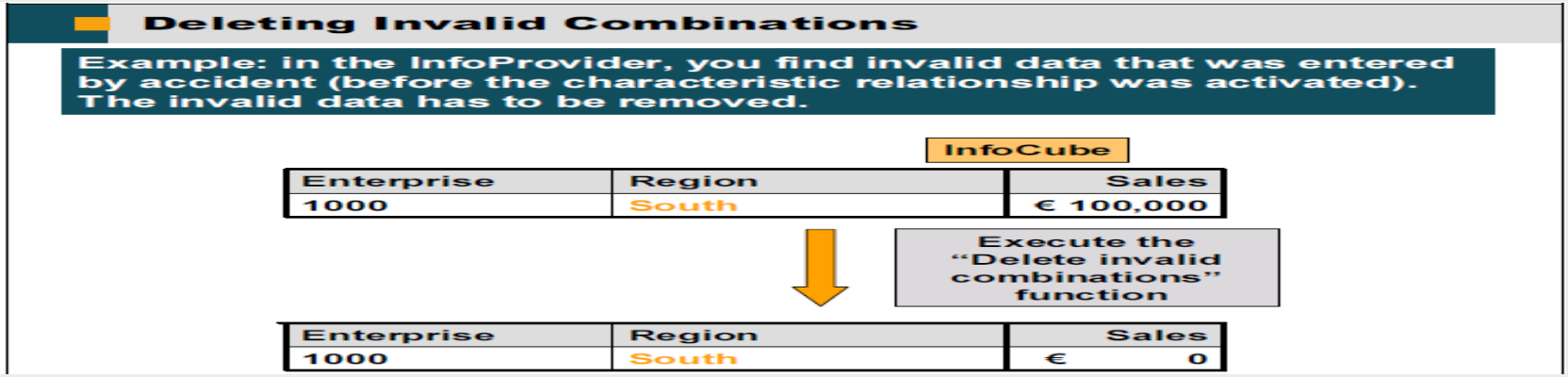
➢Derive Characteristic: we can derive values for characteristics in real time cube, we define source and target characteristics in planning modeler.

➢Combination proposal: For a given set of characteristics and a selection this service generates all valid combinations of characteristic values. This function is compatible with check combination, i.e. generated combinations are valid and there exist no valid combinations that will not be generated.

➢We have 4 types of characteristic relationships:

1. Master data attributes.
2. Hierarchy
3. Data store object
4. Exit class.

Characteristic Relationship



- The following errors may occur with characteristic relationships: changes in the data modeling area can lead to invalid data (particularly if characteristic relationships were added after going live).
- You can use the "Delete invalid combinations" planning function to delete the key figure values for all the records that do not match the characteristic relationship.
- The above figure shows the data display from the query viewpoint.
- You have to comply with two technical prerequisites if you want to use this planning function:
  - It can only be created on an aggregation level that refers directly to a real time Cube; in other words, you cannot use it if the aggregation level is based on a Multiprovider.
  - The aggregation level has to contain all the InfoObjects from the real-time Infoprovider that are valid in the aggregation levels.

Integrated Planning – Modeling

- Real time InfoProviders
- Aggregation Levels
- Filters
- Planning Functions
- Planning Sequences
- Characteristics Relationship
- **Data Slices**
- Input Ready Queries

Data Slices

- The data slice is a concept for protecting the main data of a planning-relevant InfoProvider against changes. This protection affects all input-ready queries and planning functions that use this planning-relevant InfoProvider.

- Example : If you want to ensure that certain plan versions cannot be changed after a certain point in time for example, and prevent current data from being overwritten, you can use a data slice containing these plan versions.

- Data slices are created for a real-time InfoCube or a DataStore object for direct updates in planning mode. They affect all planning InfoProviders that contain this planning-relevant InfoProvider.

Data Slices

There are two types of data slice:

- Data slices based on a selection. Here, you set the restrictions for the characteristics that you want to protect against changes.

- Data slices based on an exit class. In the exit class, you can implement a customer-specific logic to protect data records.

- The following basic rules apply for data slices:

  If no data slice is defined for a planning-relevant InfoProvider, any valid characteristic combination can be posted to this planning-relevant InfoProvider.

Data Slices

Every data record in the selection in a data slice is protected against changes. The corresponding cells in input-ready queries cannot be changed. You cannot use planning functions to change or save this kind of record. The data slices cumulate.

If a real-time enabled InfoProvider contains a data slice without any characteristic value restrictions, the data slice acts as a lock for postings of all types in the entire real-time InfoProvider.

Once you have created a data slice it is activated automatically. The settings in the data slice definition have an immediate effect on the ability to update data. You can deactivate an existing data slice at any time (status inactive). This data slice is then ignored.

Data Slices

➢Defining Data slices (Procedure):

The display mode provides you with an overview of the available data slices. In change mode, you can change data slices and define new data slices:

To create a data slice, choose Create New Data Slice. The row of the data slice to be created is selected.

In the Description field, enter a text for the new data slice.

Specify whether the data slice should be based on a selection or an exit class.

If the data slice is based on a selection, set the required characteristic values in the lower screen area:

Select the characteristic that you want to restrict.

Select the symbol in the Values column. The dialog box for specifying the characteristic restriction appears.

In the value list, select one or more values. The selection can also contain variables, provided that the variables do not send any dialogs at runtime.

Choose Add and save your selection by pressing OK.

Data Slices

- If you want the data slice to be based on an exit class, enter the name of the exit class.
- Under Restricted, choose the characteristics that you need in the exit. You will only get the current values for these characteristics in the exit. If you are also interested in the initial values in the exit, set the flag in the also # column. If this flag is not set for a characteristic, the exit is not called for aggregation levels that do not contain this characteristic. The characteristic value would be initial in this case.
- The exit class must implement interface 'IF_RSPLS_DS_EXIT'. Only these types of classes are offered for editing in maintenance. We recommend that the customer class inherits from template class 'CL_RSPLS_DS_EXIT_BASE'. The template class itself can be run directly, but does not execute an action. Re-implement method 'IS_PROTECTED'. Note the example source code given in the template class too: An infrastructure for buffering can be provided here.
- If you want the data slice to be initially inactive, remove the flag from the Active field.

Integrated Planning – Modeling

- Real time InfoProviders
- Aggregation Levels
- Filters
- Planning Functions
- Planning Sequences
- Characteristics Relationship
- Data Slices
- **Input Ready Queries**

Input ready queries: the base for input templates

You can use input-ready queries to create applications for manual planning. These applications range from simple data entry to complex planning applications.

You define an input-ready query in the **BEX Query Designer**.

Then, in the **Web Application Designer** or the **BEX Analyzer**, you can combine the queries with other queries and planning functions to create complex planning applications, as required.

You can use the following **InfoProviders** to define an input-ready query:

- Aggregation levels
- Multiproviders, containing at least one simple aggregation level

As a prerequisite for manual planning, the real-time InfoProvider that you are using for planning must be set to "planning enabled".

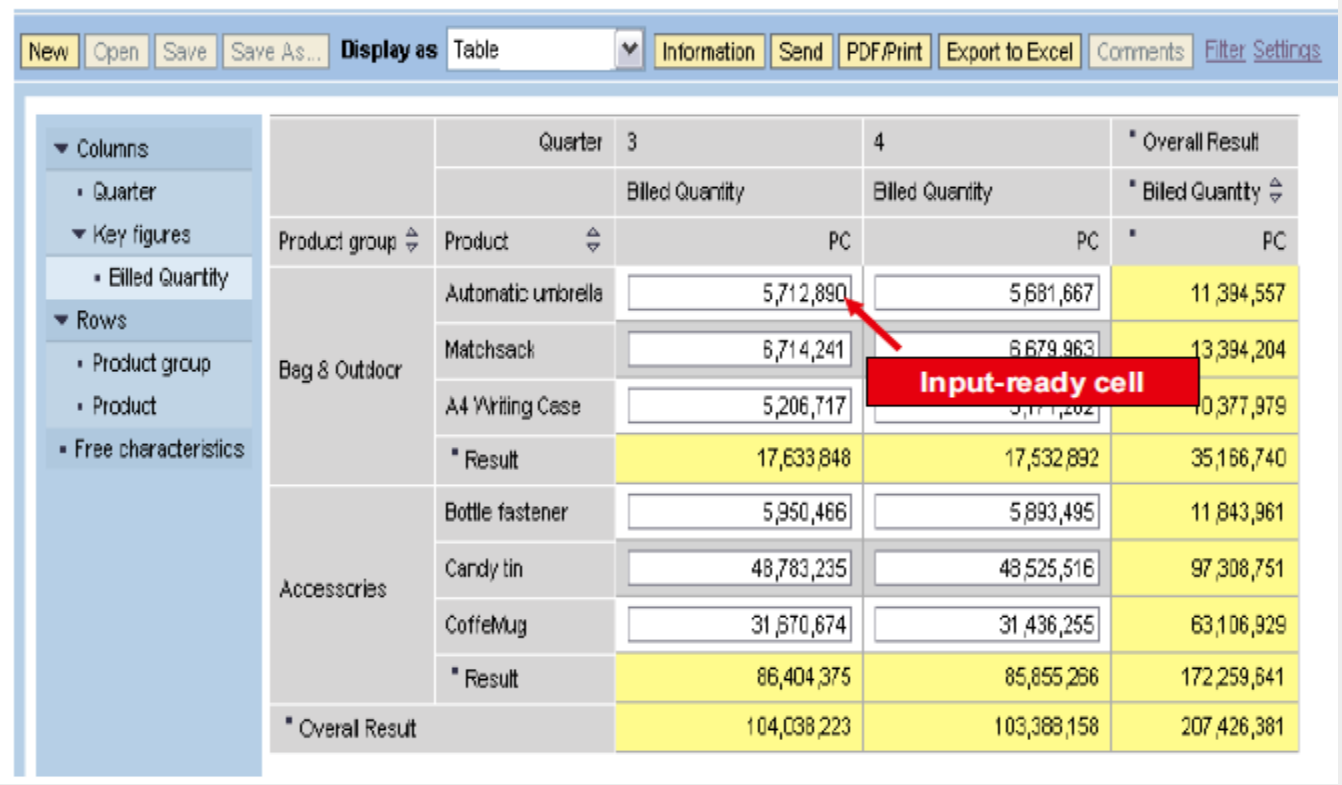
The new **BEx Query Designer** is used to set up the **input-ready queries** for manual planning.
This means that the query S properties and settings (as shown above) are also available for planning.

Note that any number of characteristics may be included in the Free characteristics area and a maximum of only 50 "Number of structures" characteristics can be selected in the Rous and columns area

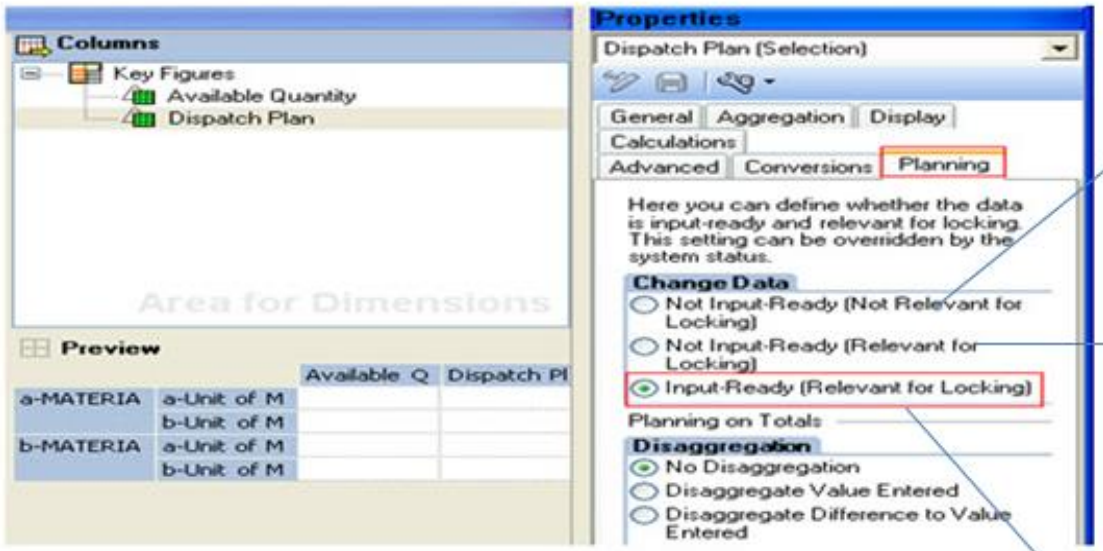
So if the query contains no structure, there can be a total of 50 characteristics in the rows and columns.

If you use a structure, there is a total of 49 characteristics allowed in the rows and columns and if you use two structures, a total of 48 is allowed.

Input ready queries: A typical Input template



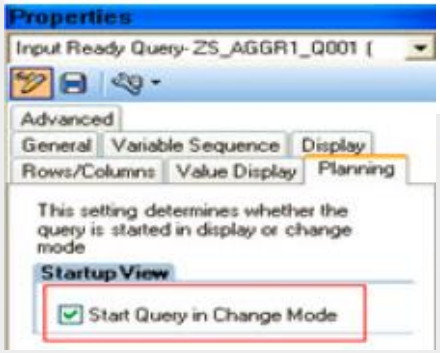
Input ready queries: Settings in query designer



The structure components are **not** locked for the exclusive access of a user because many users use this data as a reference (for actual data).
This is the default setting.

If you want to protect structural components against manual entries but allow changes by planning functions, you can use locks to protect this data for one particular user. This allows you to ensure that the planning function works with the displayed data only and not with data that has been changed by other users.

The data is locked for a user and is input-ready for manual planning.



In addition, you can specify whether an input-ready query is started in **change mode** or in **display mode**. You can find this property in the ***Query properties*** on the *Planning* tab page.

Input ready status in a query

General requirements for input-ready cells in a plan query

1. Real-time InfoProvider forms the basis of planning.
2. The "can be planned" load behavior is set for the real-time InfoProvider
3. The plan query is based directly or indirectly on aggregation levels.
4. The input-ready cell must be filled from the aggregation level.
5. "Start query in change mode" must be selected in the plan query.
6. Cells in the plan query must be set to the "planning" property.
7. There must be no locks.
8. There must be no data slices set.
9. The aim of planning must not be a combination not allowed by the characteristic relationships.

Input ready status in a query

Each characteristic in the aggregation level must have exactly one single value for the input-ready cell.

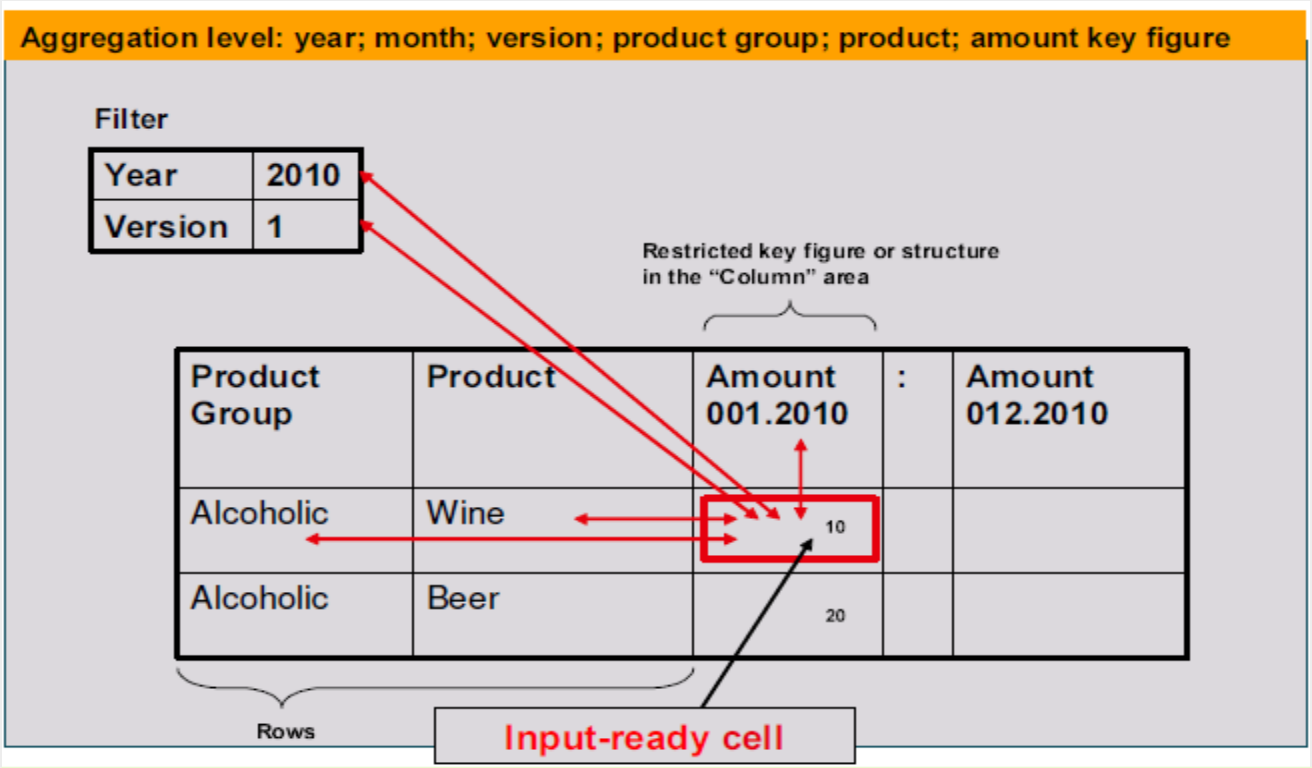
A simple method for obtaining an input-ready query is to first include all the characteristics from the underlying aggregation level in the drilldown (rows or columns). This action immediately supplies you with the input-ready cells. You could position all the characteristics in the rows and all the key figures in the data columns.

If the query is then ready for input, you can remove single characteristics one by Ole from tie drilldown and include them in the filter, or use structures or restricted key figures in the data columns instead of key figures.

If the query is not ready for input after this, then you should use the checklist of the general prerequisite criteria for an input-ready query (as mentioned before) to find the error.

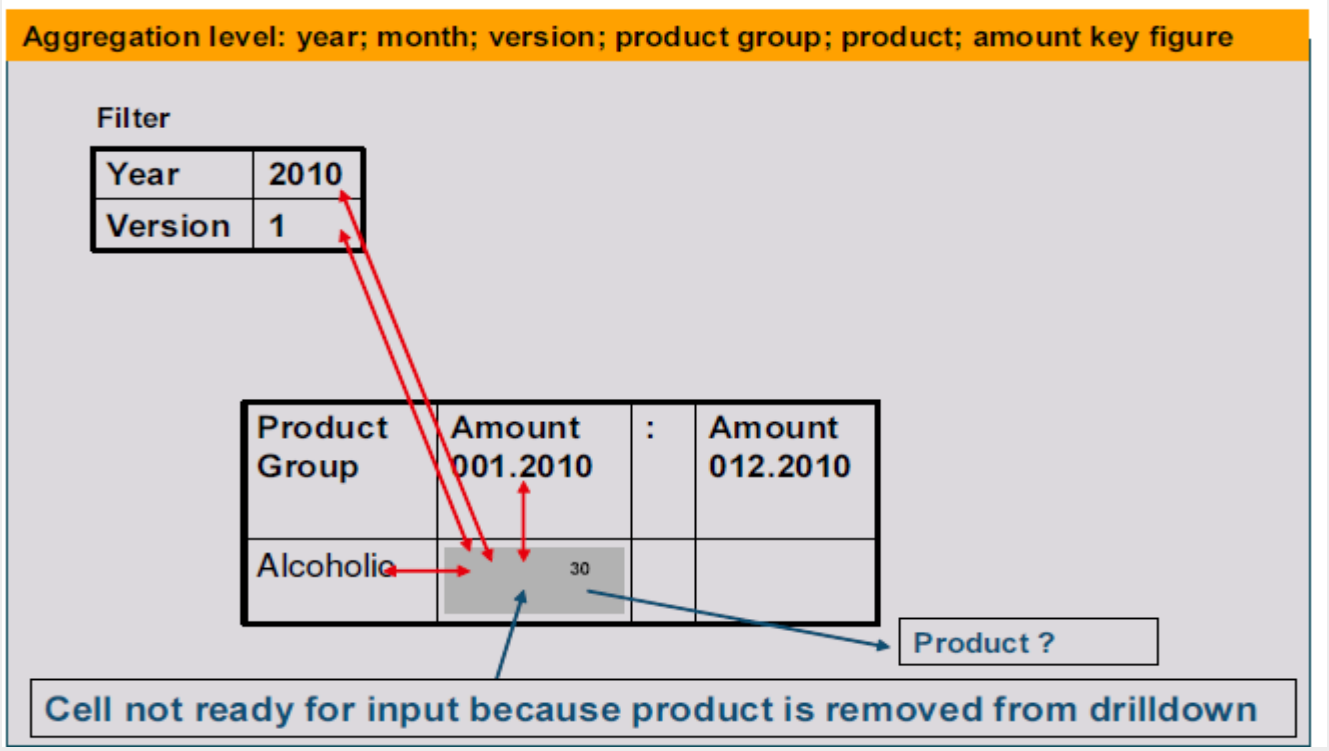
Essentially, your aim is to **set the plan query to the finest granularity of aggregation level in order to receive an input-ready cell and thus be able to plan in a query.**

What causes the input ready status



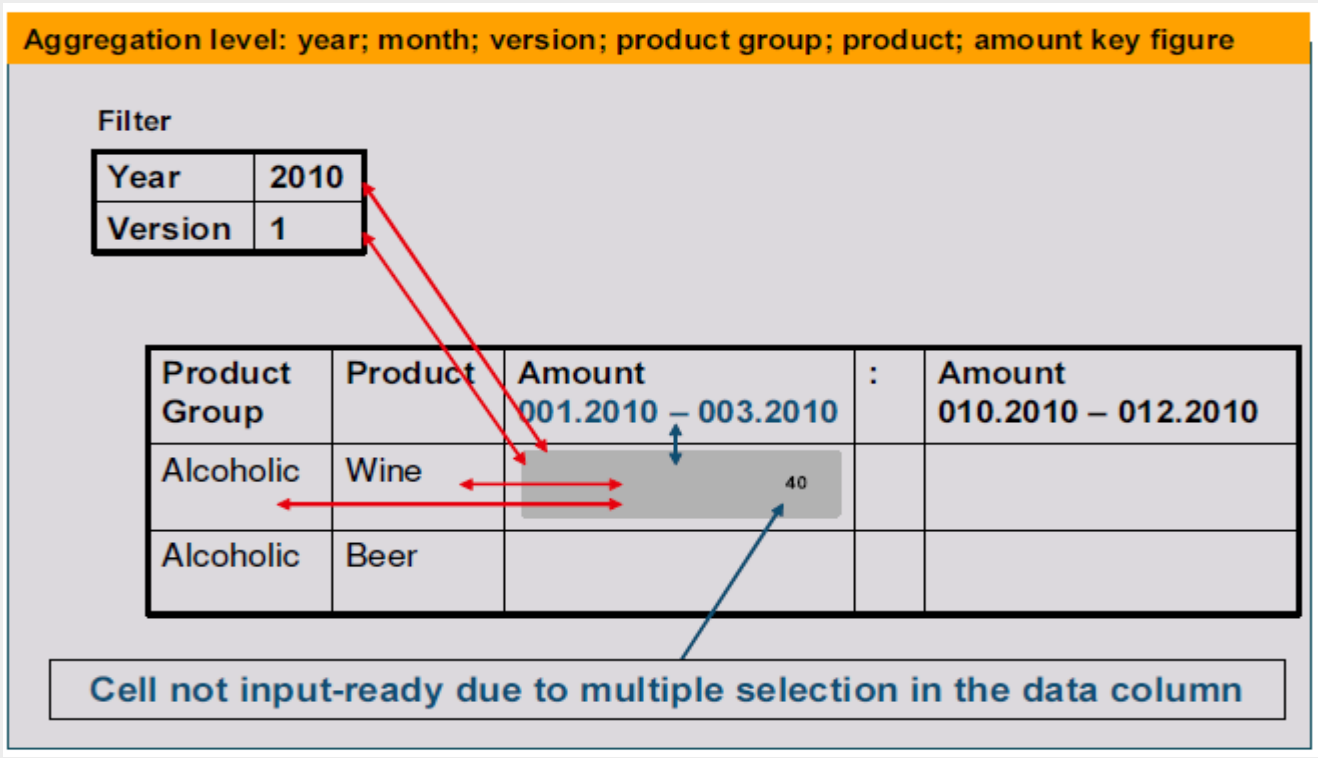
The above illustration shows that the finest granularity leads to the input-ready status in the plan query, because DOO the characteristics contained in the aggregation level are included in either the drilldown or the filter, and there is no multiple selection of the characteristics in the data columns or in the filter.

When Input ready status not occur – Case 1



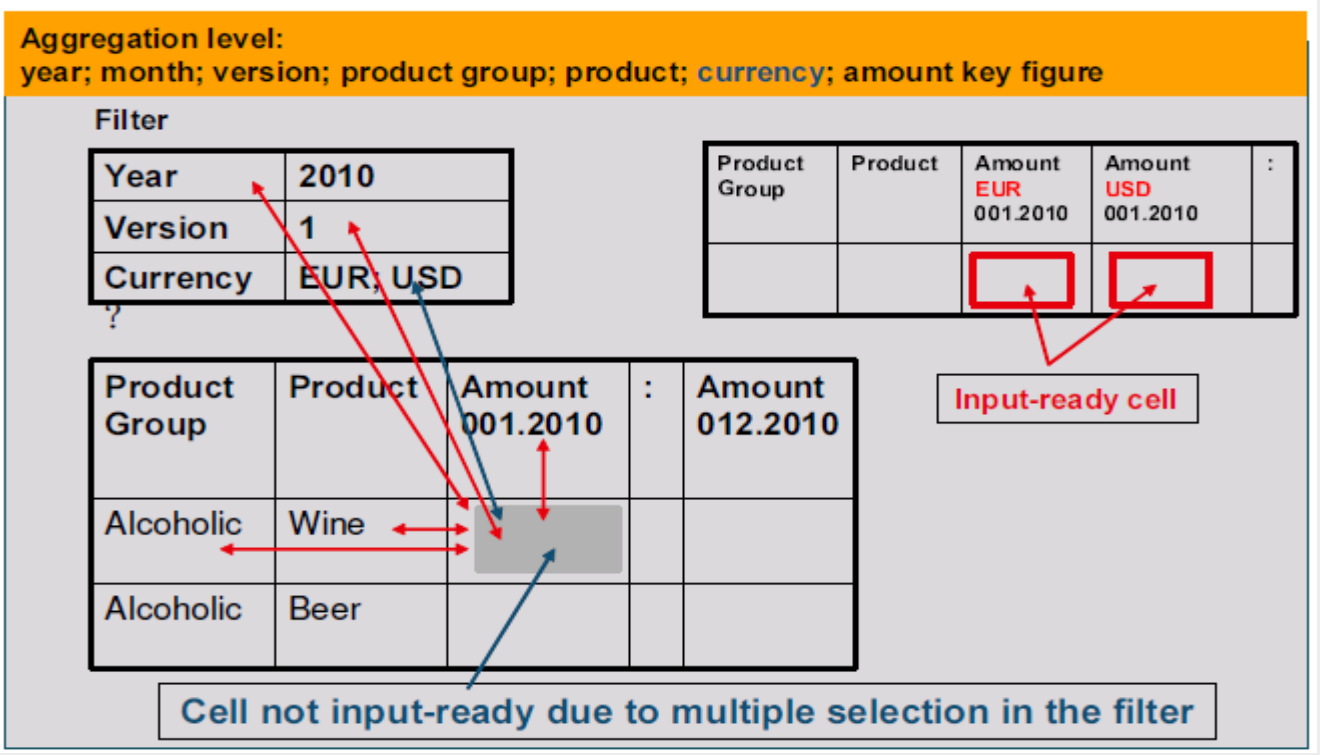
In the illustration above, the product characteristic contained in the aggregation level on which the query is based is not included in either the drilldown or the filter. This means that the data records are totaled using the products, the finest granularity is not reached, meaning that no input-ready cell results.

When Input ready status not occur – Case 2



In the example above, you can see a cumulated display over three months in the data columns, where there is no single value per column, resulting in a summation. The cell that is used to cumulate over the months is therefore not ready for input.

When Input ready status not occur – Case 3



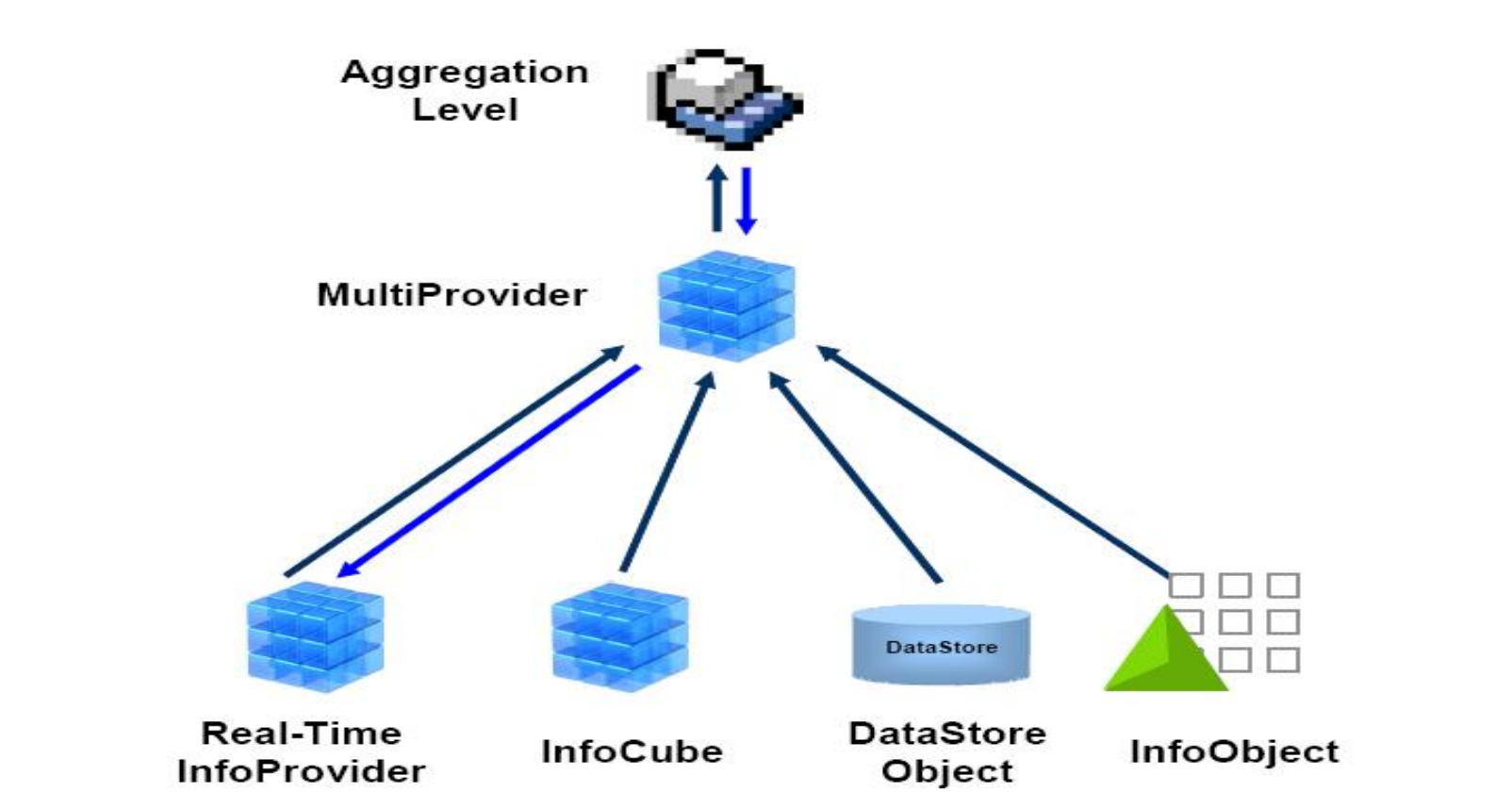
if more than one currency is involved, the different currencies are selected as single values in separate data columns, to enable write access. If the selection in the filter is set to multiple selection, there is no input-ready status accordingly.

Input ready plan query will input ready if:

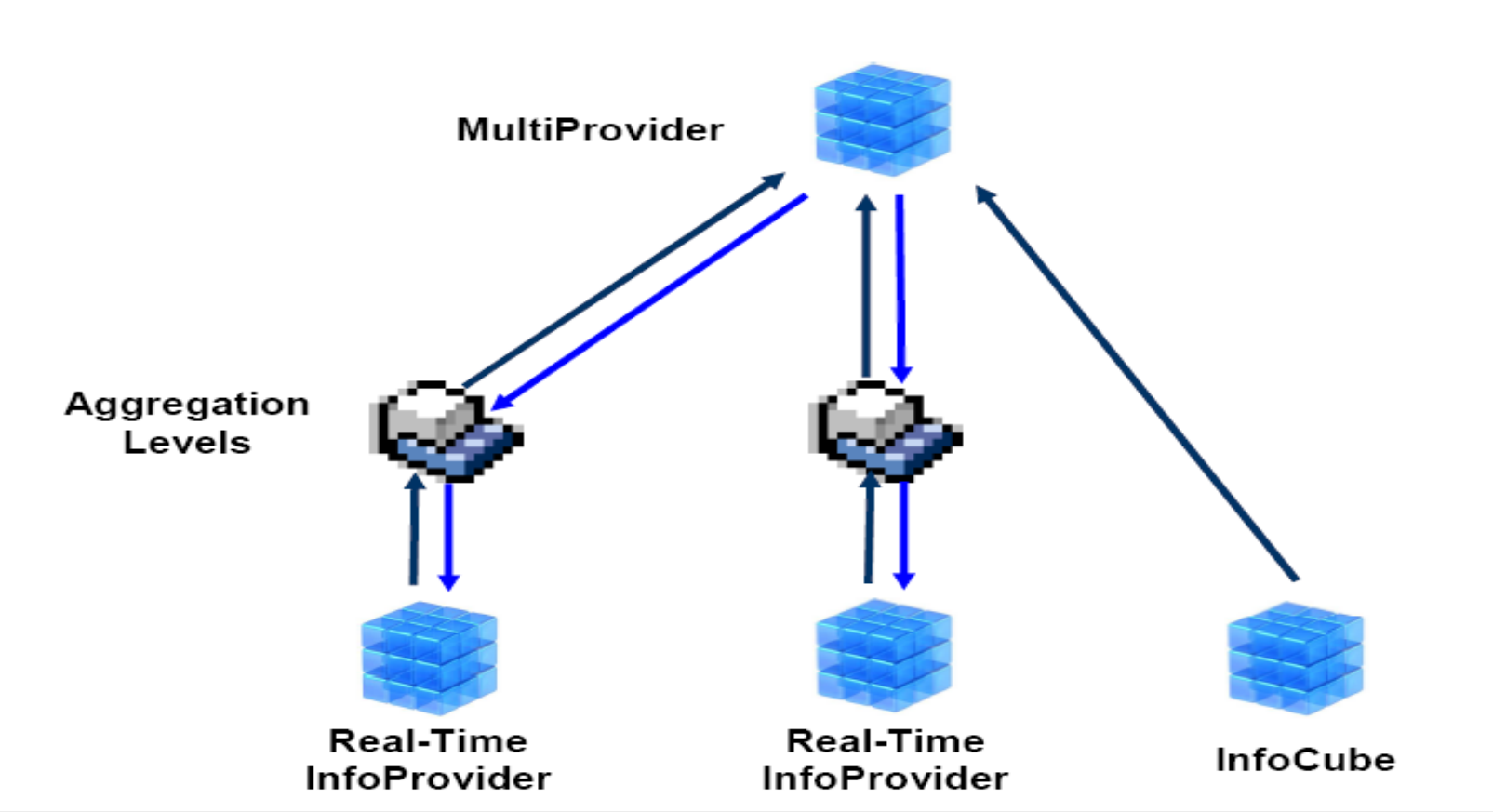
- The filter changes in the executed query.
- Characteristics are moved from the drilldown to the filter.
- The sequence of columns is re-sorted.
- Rows and columns are swapped.

Result:
A plan query remains ready for input if, even despite any changes made at runtime, each characteristic in the aggregation level is restricted to a single value.

Creating an Input-read Query (1)



Creating an Input-read Query (2)



Capgemini

CONSULTING.TECHNOLOGY.OUTSOURCING

People matter, results count.

About Capgemini

With more than 140,000 people in 40 countries, Capgemini is one of the world's foremost providers of consulting, technology and outsourcing services. The Group reported 2014 global revenues of EUR 10.5 billion.

Together with its clients, Capgemini creates and delivers business and technology solutions that fit their needs and drive the results they want. A deeply multicultural organization, Capgemini has developed its own way of working, the Collaborative Business Experience™, and draws on Rightshore®, its worldwide delivery model.

Learn more about us at www.capgemini.com

Rightshore® is a trademark belonging to Capgemini

www.capgemini.com