SAP BW Data Dimension Modeling

Lonnie D. Ayers, PMP
In this Guide, We Cover Some of Our Modeling Guidelines for InfoCube Dimensions-Part II

www.SAPBWConsulting.com
Introduction to Data Modeling ................................................................. 4

Traditional Entity Relationship Modeling ............................................. 4

SAP BW Dimension Design .................................................................... 8
  First Pass At Dimensional Modeling .................................................... 9
  Compound keys .................................................................................. 16
  Combining Dimensions ..................................................................... 24
  Broad Dimensions ............................................................................ 24

Optimal Fact Table Size ...................................................................... 25
  Line Item Dimensions ..................................................................... 25

Other Performance Enhancement Techniques ....................................... 25
  Aggregates ....................................................................................... 26
  Demographic Characteristics ............................................................. 26
  Categorization During Query Runtime ............................................ 26
  Time Dependency as Categorization Characteristic ....................... 26
  Dealing With Historical Data Changes AKA Slowly Changing Dimensions .... 27

Additional Categorization Techniques ................................................. 27
  Partitioning ...................................................................................... 29
  Degenerated Dimensions .................................................................. 33

SAP Extended Star Schema Diagram ................................................. 35

SAP BW Consulting Data Modeling Best Practices ............................ 36
  Key Figure Based Approach ............................................................ 37
  Bottom-Up Scenario ......................................................................... 39

Determining Data Flows ..................................................................... 41

Modeling SAP BW Multi-Providers .................................................... 42

SAP BW Consulting Data Modeling Value Proposition ....................... 44

Figure 1 Preliminary InfoCube Design ................................................. Error! Bookmark not defined.
Figure 2 Aircraft Structure Sample-F-4 Phantom .................................... 9
Figure 3 Technical Structure ................................................................ 10
Figure 4 Relationship of Characteristics in Dimension Tables .................. 10
Figure 5 M:N Relationship of Master Data ............................................. 10
Figure 6 M:N Relationships: Two Dimensions ..................................... 11
Figure 7 M:N Relationship: One Dimension ........................................ 12
Figure 8 Use SE38 to see SAP_IINFOCUBE_DESIGN .......................... 15
Figure 9 Examine InfoCube for Line Item Applicability ......................... 15
Figure 10 Compound Master Keys ........................................................ 14
Figure 11 MultiProvider Overview ...................................................... 16
Figure 12 Additional Modeling Features .............................................. 17
Figure 13 More Modeling Features ...................................................... 20
Figure 14 Unrelated Characteristics in the Same Dimension ............................................. 21
Figure 15 Mixed Dimension (1) ............................................................................................ 22
Figure 16 Mixed Dimension (2) ............................................................................................ 23
Figure 17 Broad Dimensions ............................................................................................... 25
Figure 18 Category Dimensions .......................................................................................... 27
Figure 19 Category Characteristic ...................................................................................... 28
Figure 20 Partitioning: Modeling Features ......................................................................... 26
Figure 21 Partitioning Example (1) ..................................................................................... 31
Figure 22 Partitioning Example (2) ..................................................................................... 32
Figure 23 Degenerated Dimension or Line Item Dimensions .............................................. 33
Introduction to Data Modeling

In traditional Entity Relationship Modeling, the core concept is very simple. It essentially boils down to the fact that at an elementary level, a thing, say a product, is related to another thing, say a customer. The entities, Product and Customer, in this case, are related in some way. The relationship is the key point, as it can be one-to-one, one-to-many, or many-to-many and is traditionally denoted with the following symbols.

![One-to-One Diagram](image1)

![One-to-Many Diagram](image2)

![Many-to-Many Diagram](image3)

To make these relationships mean something to the two entities, you need a key, or a way to link the two. As we will discuss later, keys are everything!

Traditional Entity Relationship Modeling

In his book, Case*Method: Entity Relationship Modeling, by Richard Barker, 1990, the subject of Entity Relationship Modeling is dealt with at length. Not much has changed since this book was written, but I believe for an even more in-depth understanding you would be well served by reading the book, “Data Model Patterns, Conventions of Thought” by David C. Hay. In this second book, you will find data model patterns for virtually any business scenario you are liable to run across.

---

www.SAPBWConsulting.com
E-Mail: Lonnie.Ayers@SAPBWConsulting.com
Share on FaceBook       Post on LinkedIn       Tweet This Guide
The diagram above is meant only to give you an idea of what an Entity Relationship Diagram looks like. In a full sized system design, the ERD may have hundreds if not thousands of entities all connected in a similar fashion.

Without attempting to provide a complete treatment on Entity Relationship Modeling, I would like to just make you aware of the following five points:

1. The existence of Many-to-Many, denoted as M:M, usually indicates the existence of an unidentified intersection entity. In the example above, this missing entity might well be Purchase Order, which would have both header and detail section entities.
2. There exist a concept called Normalization, which ranges from 1st degree to 5th degree of normalization. There are certain modeling precepts that dictate which level you must take the normalization too.
3. There are many modeling concepts that drive you to intentionally introduce redundant normalization techniques.
4. When converting from the ERD model to the Star Schema, the structure of relationship modeling changes, however, one key element remains of vital importance, the identification of and proper utilization of Primary, Secondary, Compound and Foreign Key relationships. Within the realm of ERD, these keys connect the Entities. Within the Star Schema, they become DIM ID, and SID IDs, and serve the same basic purpose.

5. Within ERD models, the database management system takes care of the indexes and internal hash tables (a complete separate and complex subject itself) with far less involvement required of the DBA, although some remains necessary. Within the SAP BW Star Schema based system, the effort to keep these keys, i.e., DIM and SID keys, requires effort on the part of the functional SAP BW consultant, the SAP BW BASIS Administrator and can largely but totally be automated via proper use of process chains.

What is especially important to realize is that the use of CASE or Computer Aided System Engineering tools, of which there are several vendors, allows you to quickly model not only the Entity Relationship Diagram, but as well, the Process Flow Diagram, Function Hierarchy Diagram, and to directly generate complete functional systems from the tools. This author’s expertise was in the use in the Oracle Designer 2000 tool set, now much evolved, to generate fully functioning business systems. There currently are no such tools for developing SAP Extended Star Schema’s nor of going straight to the fully functioning system you need. There is, however, a vast library of what is referred to as SAP Business Content and a couple of minimally useful tools available from SAP to navigate the SAP Business Content. This is why you must have SAP BW consultants who not only know the SAP BW Tool, but who also have read and are knowledgeable of the contents of the SAP Business Content, and can translate from the Business Content to your Business Requirement.

With this brief background, which if you read part one of our series on Data Modeling, you will be familiar with, we now move on to Dimensional Modeling.
SAP BW Dimension Design

One of the most critical aspects of SAP BW is system performance, most typically experienced by the user as query performance response times. Whether it is fast or slow, from a user perspective, it is never fast enough! Though there are many items which you can report on in the SAP BW, called InfoProviders, one of the most influential design elements is the design of the InfoCube Dimensions. Shown below is a typical Star Schema arrangement, in this case, for an airline route profitability solution, which is not standard SAP BW Business Content but will serve our purposes to illustrate how to design efficient dimensions.

In our example above, which is not displayed in the extended Star Schema in order to facilitate communication between the Business User and the BW Consulting team (and is our Best Practice recommendation for how you should use it when interacting with your business user), you can see that we have defined the dimensions based on their semantic meaning. In laymen's terms, we have grouped characteristics according to what the user saw as related or having a relationship.
First Pass At Dimensional Modeling

In our first pass at defining the dimensional model, we first had to think about what was related in the business context at hand. In this case, Airline Route Profitability, we decided that one naturally related subject area within Airline Route Profitability was aircraft. Aircraft have a highly defined structure defined by governing agencies, such as the IATA and FAA. The military has a very similar system, and if you know how to read the aircraft designation, say F-4C Phantom, you can easily determine the specific technical specification of the aircraft. In terms of dimensional modeling this means you can define a dimension in such a way that can you assign or roll-up cost, from the lowest level. In the case of Airline Route Profitability, the lowest level is a Flight Segment, flown by a particular aircraft, on a particular date and time, with a specific set of passengers and cargo, each of whom paid a specific price and flew in a particular class, i.e., 1st, Business Class, Economy Class.

In our example below, we defined a dimension using the Aircraft All and Aircraft SubType Master Data tables. Each of these tables creates a SID or Surrogate Identification Number, which is a number present in both the external Master Data Table and the Dimension Table.

This process is replicated for each of the other Dimensions. In turn, each Dimension Table has a DIM (Dimension Identification Number) which is linked to the central Fact Table of the InfoCube.

Figure 1 Aircraft Structure Sample-F-4 Phantom

www.SAPBWConsulting.com
E-Mail: Lonnie.Ayers@SAPBWConsulting.com
Share on FaceBook  Post on LinkedIn  Tweet This Guide
In order to minimize query response time, one of the most important decisions the BW Consultant must make is how to assign the various Characteristics to the dimension tables. Ideally, the relationship is 1:N (One-to-Many or Each Mother may have one or more children but each child can have only one biological mother, though they may have many mothers).

In a dimension, for every unique combination of Characteristic SIDs, i.e., Aircraft SubType, Aircraft All, there is a unique DIMID assigned. At any given point in time, this means there is only one single DIMID assigned for that row. However, if the combination of SID Values changes at a later point in time, a new DIMID will be assigned. In this case, you have a N:M or (Many-to-Many relationship or Each Aircraft can fly one or more routes and each route can be flown by or more aircraft subtypes). These situations typically will result in large dimension tables and sluggish performance from the user perspective. You want to avoid this at all cost.

In addition to the DIMID as the key of the dimension table, there is one column for every characteristic in the dimension table. These columns contain the SID (Surrogate ID) values for each characteristic value. The actual characteristic value is the key to the SID table. These SID values are assigned to the corresponding characteristic value when master data is loaded to the master data tables. In turn,
when transaction data is loaded to the InfoCube, the appropriate SID values for the characteristic values of the transaction are placed in the dimension tables.

The maximum number of entries of in the dimension table is the cartesian product of all SIDs. For example, if there are 1000 flights and 1000 routes, the cartesian product has 1,000,000 possible entries in the dimension table.
Figure 4  M:N Relationship of Master Data

Example: Material and Color
- If COLOR is an attribute of the MATERIAL characteristic, the master data table for MATERIAL should contain COLOR in addition to MATERIAL TYPE.
- However, this is not possible because MATERIAL is the unique key for the master data table. Therefore, the master data table can only contain one material with several colors if it is time dependent. This is a typical problem with star schemas.

Variants in the Source System
(Compound Keys)

<table>
<thead>
<tr>
<th>Material</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>RED</td>
</tr>
<tr>
<td>A</td>
<td>GREEN</td>
</tr>
<tr>
<td>C</td>
<td>RED</td>
</tr>
<tr>
<td>D</td>
<td>BLUE</td>
</tr>
<tr>
<td>D</td>
<td>YELLOW</td>
</tr>
<tr>
<td>E</td>
<td>RED</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MATERIAL</th>
<th>OBJECT</th>
<th>DATE</th>
<th>DATEFROM</th>
<th>CHANGED</th>
<th>COLOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
<td>20100412</td>
<td>20030401</td>
<td>RED</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>99991231</td>
<td>20100413</td>
<td>GREEN</td>
<td></td>
</tr>
</tbody>
</table>
Although it is technically possible that all possible combinations of Characteristics could be loaded, that normally doesn't happen nor is it likely to happen. As the **BW Consultant** you need to evaluate what sort of data combinations are likely to actually be loaded during actual data loads given the Business Process Functionality and design it so that large dimension tables do not occur.

In most situations, M:N relationships mean that there is a missing intersect entity and that the two characteristics should be stored in different dimensions, for example Aircraft and Route. This sort of relationship is normally defined by facts or key figures that arise when transactions occur, i.e., flights.

In companies that use standard SAP functionality, for example, the Flight Object, which can be used to track planned and actual flight operations within the SAP Industry Solution for both Defense Forces and the Airline Industry, several types of flights are often used for every type of airplane (or voyages for ships or trucks or trains). This sort of event provides information that reflects characteristic

---

**www.SAPBWConsulting.com**  
E-Mail: Lonnie.Ayers@SAPBWConsulting.com  
Share on FaceBook  
Post on LinkedIn  
Tweet This Guide
relationships that exist at the time of transaction or event. This type of solution can help you capture both the cost of a flight, as well as whether it took off on time, landed on time, or in the case of military operations, what type of weapons it departed with and what kind it landed with.

![M:N Relationship Modeled in One Dimension Tables](image)

*Figure 6 M:N Relationship: One Dimension*

In most cases, the alternative modeling option is to model the N:M relationship in one dimension, but this will not be the best solution.

After you have built your model, and loaded some data, you should check the actual number of records loaded in the dimension table versus the number of records loaded in the fact table. The general rule of thumb is that the ratio of dimension table records to fact table records should be between 1:10 to 1:20. If you can keep your ratios within this range you will end up with a relatively small dimension table and larger fact table.

[www.SAPBWConsulting.com](http://www.SAPBWConsulting.com)
E-Mail: Lonnie.Ayers@SAPBWConsulting.com

Share on FaceBook    Post on LinkedIn    Tweet This Guide
To determine this ratio, you will need to go to Transaction SE38, and under program type, enter SAP_INFOCUBE_DESIGN and execute.

![SE38 Editor: Initial Screen](image)

**Figure 7** Use SE38 to see SAP_INFOCUBE_DESIGN

![List of cubes in the system and their layout](image)

**Figure 8** Examine InfoCube for Line Item Applicability

---

www.SAPBWConsulting.com
E-Mail: Lonnie.Ayers@SAPBWConsulting.com

Share on FaceBook    Post on LinkedIn    Tweet This Guide
Compound keys

In order to facilitate time-dependent reporting, it is sometimes necessary to use compound keys. In order to model a compound key, there is a check box for compound keys in the InfoObject Characteristic. These special attributes then form the first, or superior, portion of the compound key. The characteristic value itself forms the subordinate portion of the key. Compound keys form a single key of a characteristic during the staging process. Therefore, when data is extracted for this characteristic, values for each of the key components must be available for extraction. For both Master Data and Transactional Data Loads, this holds true.
Figure 10 MultiProvider Overview
Additional Modeling Features

➢ Place the most selective characteristic at the top of the dimension table.

➢ Do not model Characteristics with values that frequently change in relation to another “Main” Characteristic in a Dimension nor as a time-dependent attribute for the “main” characteristic
   ➢ For example, Material and Promotion Status
   ➢ This would result in large dimension tables or attribute tables.

➢ Sequence of Key Figures in the Fact Table:

   ➢ Large Data Volume Scenarios:
     ➢ When adding data to fact table:
       ➢ Load those key figures are always filled first
       ➢ Load those key figures that are only occasionally filled last

Figure 11  Additional Modeling Features
Additional Modeling Features-Pt. 2

Granularity

Granularity Influences the:
- Reporting Options
- Performance
- Data Volume
- Load time, among others

Example:
- Document Number
- Item
- Posting Date
- Customer Number
- Material Number
- Material Group

Or
- Material Number
- Customer Number
- Sales Organization

- Month
- Amount
- Quantity
- Sales Organization
- Salesperson
- Division

Figure 12  More Modeling Features
Unrelated Characteristics in the Same Dimension

Why combine unrelated characteristics?

Technical Restrictions

- 13 User-Defined Dimensions – You may require more than 13 for your model

Performance and Memory

- Improve performance by reducing the number of table joins
- Combine smaller dimensions

If you merge two small DIM tables, you get a slightly larger table, but it means that only one table has to be accessed when data is loaded or a query is executed. Of course, you would not want to place two strong (such as MATERIAL and CUSTOMER) in the same dimension table. In the majority of cases, 13 dimension tables are more than enough for your modeling needs. The rule of thumb is to group characteristics with a parent:child relationship in the same dimension. This assumes that this grouping does not include any N:M relationships. If it does, this assumes that the cartesian product of such a grouping is acceptable from a performance stand-point, as illustrated in the next two figures.
Figure 14 Mixed Dimension (1)

Characteristics with an N:M relationship can be combined in the same dimension; if the maximum number of records are clearly defined and the cartesian product of the characteristic results in only a small dimension table.
In the example above, the characteristics VERSION and COLOR have an N;M relationship. Nevertheless, they can be used in the same dimension because the maximum number of records is 3X4=12. Characteristics with a large number of values receive a separate (line item) dimension (material, customer).

**Modeling Scenario:** The “PRODUCT” characteristic has the attributes “PROMOTION CHANNEL” (TV, Radio, Newspaper, and so on) and “Sales Organization” (Hamburg, Bremen, Munich, and so on). The “Sales Organization” and “Promotion Channel” combination changes frequently. In this case, you should not model these material properties as time-dependent attributes. You should instead model them as characteristics in the InfoCube. Using this technique, you can avoid having to use large attribute tables when making your selection. The “Attributes” combination is saved in the **Dimension Tables**.
Combining Dimensions

In some cases, for performance enhancement reasons, you may want to combine two small dimensions. This combined dimension means that the table only has to be accessed once when data is loaded or a query is executed. However, you would not want to combine a Dimension that contained two strong entities, such as Aircraft and Route. Besides the 3 default Dimension Tables that each InfoCube comes with, the 13 Dimension Tables provide plenty of dimension for all but the most complex of situations. The general rule of thumb when modeling is to group characteristics that have a parent:child relationship in the same dimension. This assumes the group does not have any N:M relationships (Many-to-Many). If it does have these N:M relationship, they can generally be combined when the cartesian product of these N:M relationships result in a low, fixed number, for example, Aircraft with a limited number of Aircraft Subtypes available. When the complexity or other reporting requirements require it, you can create Multi-Providers, which will discuss shortly.

Broad Dimensions
Broad Dimensions

How do we handle Dimensions and Master Data Tables that contain millions of records?

Broad Customer Dimensions

- Use the demographic attributes of the CUSTOMER to create a smaller demographic dimensions.
  - Improves query performance significantly
  - Available immediately after transaction data is loaded
  - High demands made on the system during loading

- Use Aggregates for Demographic Characteristics in the CUSTOMER Dimension
  - Improves query performance significantly
  - Requires a large amount of memory space
  - Aggregate maintenance required after a data transfer

A combination of both methods is the most successful

Figure 16  Broad Dimensions

Optimal Fact Table Size

In general, a relationship of “10-20-1” between the size of the fact table and the size of the Dimension Table is desirable. This may or may not be precisely possible, but in general, your design should result in a small Dimension Table and Large Fact Table. If not, you will find performance gradually but continually degrading.

Line Item Dimensions

Note: There may be cases where you want to define a Line Item Dimension. A typical example is when a Retail company with a large number of customers would place the CUSTOMER characteristic in a separate dimension and make it a Line Item Dimension.

Other Performance Enhancement Techniques
Aggregates

Aggregates are really small InfoCubes, a subset of an existing InfoCube. The system can recommend aggregates and you can designate aggregates manually. A typical example would be profitability by route by region. The aggregate would have this totaled already, and this would decrease (make it faster) your query response time.

Demographic Characteristics

Demographic characteristics can be used to categorize a large number of characteristics with large value sets. Categorization means you use a small number of characteristics to group a large number of other characteristics into smaller groups. The major benefit of categorization is in reducing the amount of data that must be presented in a query result or that must be handled during data staging and storage. In SAP BW, this categorization can be done in several ways and at several times. In data staging, the logic converts a characteristic value for Master Data or Transaction Data into specific category values for storage. You can use the logic in:

- Transfer Rules
- Update Rules

Each Route flown could be converted into a Group of all flights flown by different aircraft.

Categorization During Query Runtime

You can also use categorization during query execution. The technique would be to use a categorization characteristic, like “Customer Class”, versus reporting a separate row in the report for each individual report. To do this, one method would be to use a navigation attribute or Hierarchy in the query instead of the characteristic found in the dimension table of an InfoCube.

Time Dependency as Categorization Characteristic

As the time dimension is provided by default for every InfoCube. It is therefore one of the easier ways to reduce your query result set. By reporting on a time-dependent attribute instead of the characteristic itself, data sent to the query result is restricted to just those values that valid on the key data of the query.

Suppose you wanted to reduce the number of rows in a dimension to just 3? In the example above, we create a category dimension that contains the characteristic

www.SAPBWConsulting.com
E-Mail: Lonnie.Ayers@SAPBWConsulting.com
Share on FaceBook Post on LinkedIn Tweet This Guide
Income Bracket instead of actual customer numbers. When you load transaction data, the customer number is used to derive the income bracket using a formula. You could also derive the income bracket when the customer master data was loaded and store that value as an attribute of the customer. This would significantly reduce the time required to calculate the Income Bracket for each transaction since it could be read directly from the customer number. Then, in a query, Income Bracket is used to categorize customer data into one of the three categories.

Dealing With Historical Data Changes AKA Slowly Changing Dimensions

Customers sometimes change income levels and therefore, would need to be recategorized. There are at least 2 possible ways to deal with this.

- Have income brackets as an attribute of the customer
- Store it in the Customer dimension

Additional Categorization Techniques

You can further enhance performance, sometimes, by placing categorization attributes in dimensions as characteristics themselves. This reduces the OLAP processing load because you have eliminated the load of dealing with navigation attributes.

Figure 17 Category Dimensions
Category Dimensions

Category Dimensions support reports by generating **Artificial Attributes** that classify a characteristic

- CUSTOMER can be classified by Income Bracket, Volume, etc.,
- Category Dimensions are usually related to attributes rather than to characteristics
  - Income Bracket – CUSTOMER income
  - Customer Volumes – Annual Sales, Potential Sales, AR Balance, etc.

The following factors determine whether you create Category Dimensions or hand over the categorization process to a query:

- Complexity of the categorization process
- Frequency categorization process is used

**The MDM decides whether or not to create a Category Dimension**

This also effects Tracking History?

What happens if, for example, a customer enters another Income Bracket?

<table>
<thead>
<tr>
<th>Income Bracket</th>
<th>Income Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>USD 0 to $50,000.00</td>
</tr>
<tr>
<td>B</td>
<td>USD $50,001 to $100,000</td>
</tr>
<tr>
<td>C</td>
<td>More than USD $100,000</td>
</tr>
</tbody>
</table>

**Figure 18 Category Characteristic**

A category dimension that contains the **Income Bracket** characteristic in this example instead of the actual customer number would reduce the number of rows of data in the dimension to a maximum of three rows. When you load transaction data, the customer number is used to search for the customer attribute of the Income. The **Income Bracket** value can then be derived from a formula. Another option would be to derive the **Income Bracket** value when the customer master data was loaded and store the value as an attribute of customer. This would significantly reduce the time required to calculate the Income Bracket for each transaction since it could be read directly from the customer number. In a query, the Income Bracket is used to categorize customer data into one of the three categories.
If the categorization changes, you need to ask how the historical data is to be represented after the change (slow-changing dimension). From an analytical point of view, the attributes in the Category Dimension have to be stored in the Master Data Table of the Categorized Characteristic. Use Aggregates with Category Attributes to further improve query performance. If a customer moves from one income bracket category to another, how do you model this change in the history? We have already discussed tracking history. For now, it is sufficient to say that there are several options as to how a Category Characteristic could be used. Two obvious examples would be to have Income Brackets as either an attribute of CUSTOMER or in a dimension along with the CUSTOMER Characteristic. Each of these approaches would change how historical transaction data is reported.

**Partitioning**

![Partitioning: Modeling Features](image)

Figure 19 Partitioning: Modeling Features

In the example presented in the above figure, there is a Version characteristic with three values:

- Original
- 1st Quarter Forecast
- 1st Half Forecast
- Actual
- Plan
- Forecast.

The characteristic Version is in a separate dimension table. This dimension is called a **Partitioning Dimension** since it is now used to partition, or separate, values stored for the two basic facts:
- Sales
- Units

Because the values of Version are stored as master data, it is very easy to add other scenarios, for example, a quarterly forecast or monthly plans. This only involves adding the additional master data values and then providing the associated logic for determining the appropriate value for Version in the update rules for the transaction data. This solution has the effect of using an additional key field to sort fact table records. The BI Content InfoObject 0VALTYPE is frequently used for this purpose. Other examples for such characteristics are the “Plan Actual Indicator” or the “Value Type”.

In **BI Content FI/CO (Financial and Controlling) InfoCubes**, the actual data is usually saved separately to the plan data. If needed, MultiProviders can be used to provide consolidated reports.

This is an example of a modeling with several key figures in the **Fact Table**. While there is nothing inherently wrong with this approach, it is very inflexible if the categories used (**Current, Plan, Forecast**) change over time. If changes are necessary, a new **InfoCube** must be created with new **Key Figures**, and data must be reloaded or deleted from an existing **InfoCube**. Furthermore, the structure of the Fact Table must be changed and the data must be reloaded. Either way, that could be an expensive and time consuming process.
Partitioning: Example (1)

Simple **InfoCube** Without a Partitioning Characteristic

![Partitioning: Example (1)](image)

Figure 20  Partitioning Example (1)
Partitioning: Example (2)

In this example, only two key figures are used in the Fact Table. A new dimension with the partitioning characteristic VERSION is added.

Any new value for the characteristic VERSION can be used to enhance the reporting without changing the data model.

Normally, it makes sense to combine together Key Figure values across partitioning characteristic values. For example, the sum of “Actual Sales” and “Forecast Sales” is not a usable number for analysis. In order to prevent this from happening in a query, a setting can be made in the partitioning characteristic. If the value “Unique for Every Cell” has been selected, the characteristic must be restricted to only one value in every column and in every structure for all the queries. The characteristic cannot then be aggregated. Therefore, characteristic values cannot be used to aggregate the key figure since it is restricted by the characteristic.
Degenerated Dimensions

A **Degenerated Dimension** is a large dimension table that is approaching the size of the **Fact Table** as a measured by the number of rows in the tables. This scenario is typically bad for query performance since the **OLAP Processor** must join the two large tables to select the rows of data required from the **Fact Table**.

In a **Degenerated Dimension** with order numbers as members, all descriptive attributes are located in other dimensions.

In **BW**, such a **Degenerated Dimension** can be marked as a “**Line Item Dimension**”. By doing so, when activating the **InfoCube**, no actual dimension table is created. Therefore, instead of using the **DMID** for this dimension in the key structure of the **Fact Table**, the **SID** of the “**Degenerated**” dimension is then placed in the **Key Structure** of the **Fact Table** by the system with the field name “**RSSID**”. This can be viewed using transaction **SE11** for the table */BIC/F<InfoCube Name>*.

---

**Figure 22  Degenerated Dimension or Line Item Dimensions**
This eliminates a **Table Join** and thus improves query performance. If only a limited history of **Line Item Detail** is required, a **Data Store Object (DSO)** should be considered as an alternative to the **InfoCube**.

The setting for **Line Item Dimensions** is set by choosing the “**Dimensions**” icon on the “**Characteristics**” tab of **InfoCube** maintenance. This must be done in the **InfoCube** before any data is loaded into the **InfoCube**. On the **Define Tab** for creating dimensions, there is a checkbox for marking a dimension as a **Line Item Dimension**.

Dimensions set as **Line Item Dimensions** can only contain one characteristic. This is because there must be a 1:1 relationship between the **SID** value of the characteristic and the **DIMID**. If there were more than one characteristic, then this 1:1 relationship would not exist and it would be impossible to use the **SID** of the characteristic in place of the **DIMID** in the **Fact Table** key. It is recommended that you use DSOs, where possible, instead of **InfoCubes** for **Line Items**. Another setting available adjacent to the **Line Item Dimension** setting is the **High Cardinality** setting. Dimensions containing many unique values are said to have a **High Cardinality**. In such cases, certain types of **Dimension Indexes** are easier to maintain than others. The “**High Cardinality**” setting is used to change the method for dimension indexing in **Oracle Databases Only**.

- The indicator defines which type of Index should be used:
  - **Low cardinality** (**Bitmap Index** is created where dimension values are often repeated)
  - **High cardinality** (**B-Tree Index** are created where values are not often repeated)

Switch on the flag if the dimension size is >10% of the **Fact Table** size.

Use the option in conjunction with the **Line Item Dimension**

You have now seen how to model and optimize various aspects of SAP BW **InfoCube Dimensions**. To wrap this up, we now provide you with an extended overview of the **SAP Extended Star Schema Design** and some of our **SAP BW Data Modeling Best Practices**.
SAP Extended Star Schema Diagram

What is most significant about this approach is that data is no longer stored in the Dimension Tables. Instead, Master Data and attributes of Master Data, such as Customer and Customer Addresses, are stored in external tables and linked to the Dimension tables via SID or Surrogate Identification Numbers and the Dimensions are linked to the Fact Table via DIM ID or Dimension ID Numbers.

This approach offers several benefits, the most important of which is that NO data other than keys is stored in dimension tables. This means you only have to load and store data, i.e., customer name and address, once and it can be used in other Infocubes or Infoproviders. With an estimated loss to the U.S economy of more than $600,000,000,000.00 (yes, billion) from bad data, this is a critical improvement and one of several methods being used to reduce data errors.

The challenge facing all SAP BW Data Modelers is how to transform the entity relationship model to the SAP Extended Star Schema. The method SAP recommends is to transform the ERD into the Bubble Model, as shown in the next diagram. In this case, I have used a model of the Airline Route Profitability model, which is not part of the standard SAP Business Content but is of critical importance to the Airline Industry! It is also applicable to the Transportation Industry in general, including 3PL (3rd Party Shippers), Railroad operators, Ships, Ferry’s and Bus companies. For the purpose of introducing SAP BW Consulting Extended Star Schema Best Practices, it will serve our purposes just fine.
Airline Route Profitability Bubble Diagram

In our Bubble Diagram above, we have seven dimensions defined around a central fact table related to Route Profitability. For the purposes of clarity, we have not shown that the Master Data, say, for Fleet, is stored in external Master Data Tables.

Special Note to Experienced Oracle Entity Relationship Modelers: You may recognize that the typical scenario used for teaching ERD is the airline reservation system. Please be aware that this model, Airline Route Profitability, is not the same model although the principles are the same. In fact, you will not find a generally available public model of the Airline Route Profitability model as each airline does this in unique ways and consider it their most critical measurement of merit.

SAP BW Consulting Data Modeling Best Practices

There are two basic approaches to SAP BW Data Modeling:

www.SAPBWConsulting.com
E-Mail: Lonnie.Ayers@SAPBWConsulting.com
Share on Facebook Post on LinkedIn Tweet This Guide
1. The Top Down Key Figure Based Approach  
2. The Bottom Up Data Source Approach.

The main approach you should focus on using is the Top Down Key Figure based approach, especially when performing your Business Content check.

<table>
<thead>
<tr>
<th>Key Figure Based Approach (Top Down)</th>
<th>Data Source Approach (Bottom Up)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get a clear understanding of your logical data model</td>
<td>Focus on the Business Process of your system</td>
</tr>
<tr>
<td>Break down the granularity of your performance indicators to basic key figures</td>
<td>Check your Datasources for particular measures (KPIs)</td>
</tr>
<tr>
<td>Find and compare your base key figures with the Business Content Repository</td>
<td>Look up DataSource for Characteristics defined in dimensions in your logical data model</td>
</tr>
<tr>
<td>Compare the Scenarios of your logical data model with the Business Content Infocubes, Queries and Workbooks</td>
<td>Understand how Business Content maps to the fields you found to InfoObjects</td>
</tr>
<tr>
<td>Check for performance indicators (KPIs) in Business Content Queries</td>
<td></td>
</tr>
<tr>
<td>Investigate the data flow for identified key figures</td>
<td></td>
</tr>
</tbody>
</table>

**Key Figure Based Approach**

The Key Figure based approach looks like the below diagram in process flow form.

---

www.SAPBWConsulting.com  
E-Mail: Lonnie.Ayers@SAPBWConsulting.com  
Share on FaceBook  
Post on LinkedIn  
Tweet This Guide
The main approach during the Business Content check should be a top-down approach. If the search for corresponding Key Figures is not successful, a bottom-up approach in the special case may be the best solution.

- You must understand the business concept behind this model. Technical descriptions will never reveal a 100% solution.
- Key Figures (also known as Performance Indicators or measures, depending on the culture you are working in), may be available in the business content, and you will normally find them modeled as Calculated Key Figures. If you do not find a KPI that exactly matches the KPI you are looking for, it may only mean that either:
  1. It is available but is called something else
  2. The components needed to create your KPIs may be located in separate pieces within your Business Content or are located in Multi-providers.

A. As you can see, this indicates and validates why you must have spent a considerable amount of time understanding your own business and the SAP Business Content in order to develop a valid model!

- There is a SAP Business Content Browser available in the SAP BW Business Content, but it is normally not installed nor available, unfortunately. All of the publicly available business content is located at help.sap.com under SAP Netweaver. It is no longer available in one spot nor as a separate off-line pdf document, so it
requires a great deal of time to find, study and employ in your environment.

- When analyzing SAP Business Content, focus on the following issues:
  1. Business Context to compare Business Scenarios. Often times, unrelated business areas, have models that, at heart, look the same and can be leveraged.
  2. Check Compounding, Dimensions and the availability of Key Figures. For instance, Key Figures required to compute your KPIs may be available in Multi-Providers or could be modeled that way. Investigate whether Dimensions within Infocubes can be redefined as navigational attributes.
- Verify that Business Content KPIs are calculated the same way your KPIs need to be calculated.
- Investigate and understand all provided dataflows and extractors. Extractors may need to be extended to provide all the data elements required for your scenario.

**Bottom-Up Scenario**

The Bottom-Up or DataSource based approach looks like the below diagram. The use of the bottom up approach mainly depends on whether you are running a SAP R/3 driven system (meaning most if not all of your source data will be coming from one or more SAP R/3 systems)
1. Assumption: Project is based primarily on SAP R/3 or R/3 is at least part of the sourced systems. If either of the two conditions are the case, then the bottom up approach is recommended.

2. The major advantage of this approach is that the customer already speaks or should be vaguely fluent in SAPenese. What is SAPenese? Well, it is the specialized language of the SAP software system, which sounds like space aliens talking when outsiders hear it. Like any specialty, such as aviation, a unique language evolves and SAPenese is the result. For the SAP BW consultant, this greatly facilitates conversations about business requirements and allows you to find relative SAP Business Content far easier than in the Top Down approach.

3. Once you have analyzed the SAP DataSources, meaning you have taken a look at, for example, how Purchase Order number ranges are defined in the Material Management IMG (Implementation Management Guide), you can find corresponding Infosources (and extend them if necessary to support your scenario).

4. There is a nearly 100% chance your system environment will not be purely SAP. However, if you can find what you need in SAP and find the corresponding infosource equivalents in the other systems, it is far easier to match the datatypes and get them into SAP BW. If you are in a mixed Oracle and SAP environment, there are several third party vendors who have done the mapping and matching for you and we recommend you investigate their usage, budget permitting.
Determining Data Flows

Your data model will determine your required dataflow. Once modeled, you can always view your dataflow using the metadata repository browser, which provides you a snapshot of your end-to-end dataflow.
Modeling SAP BW Multi-Providers

A SAP BW Multi-Provider is used to combine data from Infoprovider and to make it available for reporting. The Multi-Provider doesn't actually contain any data itself as it is more or less equivalent to a database view.

The design of effective Multi-Providers is critical to success with the SAP Business Warehouse. During the Requirements Analysis phase of your SAP BW project, you must decide upon the following design elements:

- **Granularity of the Data**
  - Time Granularity (This can range from seconds to years)
  - Aggregated or detailed data
  - Snapshot/Inventory

- **Data Volume**
  - Number of Documents
  - Number of Products
  - Number of Years to report on (History)

**Required Analysis Features**

[www.SAPBWConsulting.com](http://www.SAPBWConsulting.com)
E-Mail: [Lonnie.Ayers@SAPBWConsulting.com](mailto:Lonnie.Ayers@SAPBWConsulting.com)

Share on FaceBook    Post on LinkedIn    Tweet This Guide
- Analysis Method
- Multi-dimensional or Flat File Reporting (or a combination of both)
- Required Response Times
- Urgency
- Real-Time or some lag time allowed

The above diagram shows how you can connect virtually anytime of infoprovider into a multi-provider. Each particular analytic requirement will involve slightly different connections among the various providers. The secret is to provide what the user needs without bogging down the system with too much data.
SAP BW Consulting Data Modeling Value Proposition

This is part one of a series of How-To Guides we provide on SAP BW Data Modeling. We will provide a complete treatment of the subject in a series of upcoming How-To guides. This is one of the most critical areas of knowledge for the SAP BW consultant. From a customer value perspective, it is THE MOST CRITICAL ASPECT of your BW system.

Without a solid, robust data-model, you will not be able to build or use the query's your business users need.

As Senior Level SAP Business Warehouse Consultants, we are frequently confronted with the following questions from recruiters which is why we believe you should be using our services.

Questions which indicate lack of knowledge and for which only deep SAP BW expertise can answer.

1. Do you work on the front-end or back-end?
   a. If the Data-model is not designed properly to support your current analysis requirements, by default, the SAP BW Consultant has to work on both the front-end and back-end of the system. From a purely
technical standpoint, there is no such thing as front and back end. There are only components that provide various required functionalities.

2. Do you specialize in a particular SAP BW-SAP functional area? Translation: Do you know SAP Financials, SAP Logistics, SAP Human Resources+SAP BW.
   a. SAP does not provide any training courses for SAP BW-FICO or any other area of SAP BW + SAP R/3 functional knowledge. A SAP BW Consultant, if he claims knowledge of one of the SAP Functional Areas plus SAP BW got it in one of the following ways only:
      i. Was a SAP Functional Consultant in one of the SAP Modules, such as MM or SD or FI/CO. Then he decided to become a SAP BW consultant.
      ii. The SAP BW Consultant worked on an implementation project on the business side in one of the functional areas roughly related to say, finance or logistics.
      iii. Gained some knowledge of the SAP functional module, such as SAP Finance or SAP Human Resources, from working as a SAP BW consultant on a project where he happened to work on that area. In these cases, the functional knowledge will be limited to only those functional areas covered during that particular project.

3. Do you specialize in a SAP BW Industry specialty?
   a. SAP does not provide any SAP BW Industry Training nor does it provide SAP BW Content for all SAP Industry Solutions.
      i. A SAP BW Consultant can only have gotten SAP BW Industry experience in one of the following ways:
         1. Worked on a SAP BW Industry Solution implementation, such as Airline MRO operations and learned some of it, on-the-job.
         2. Worked in a specific industry prior to joining SAP, but then will be limited to knowledge about the particular area he worked in within that industry before becoming a SAP BW Consultant.

Our value proposition is based upon three primary principles:

1. All of our SAP BW consultants are either **SAP Certified** or are working toward it. Our philosophy is simple when it comes to SAP Certification. **SAP Certification** may not make you a good consultant, but it does indicate you have put in some time studying the available material. Being a company founded by pilots, embedded software system engineers and ex-military maintenance officers, we are, by nature, very demanding of our team.
   2. We know that the available SAP Training Material is, unfortunately, limited, of less than military class quality and usually somewhat out-of-date. To fix
that, we produce these How-To Guides. They represent our deep experience in the industry and our background in engineered software systems.

3. We are fortunate to work in an industry that continues to exhibit a high-demand for expertise like ours. We are limited in numbers with huge growth targets, and our success depends upon:
   a. Providing our customers with high-quality, high value solutions and proving the value.
   b. Creating high-performance Business Analytics Consultants by continuously recruiting, training, measuring and improving our consultant’s skills.

We also know that in today’s ever more constrained budgetary environment, we must provide our clients with the means to estimate what their BW project will cost and what benefit(s) it will provide the business. To that end, we have developed two tools, the SAP BW Resource Calculator.

The BW Resource Estimation Calculator is available for download [here](#), and we encourage you to download and give it a whirl. However, we would highly recommend you engage with us to walk you through it as the subject matter requires extensive knowledge of the SAP BW product.

The other calculator we provide focuses on the use of SAP BW to enable **Through-put Accounting capabilities**, which is something unique to SAP BW Consulting, inc.