

Requirements For Modeling Dimensional Hierarchies

When you model a hierarchy in a dimension, your design must meet a few requirements to ensure that you do not receive unexpected query results.

- ▶ Key values must uniquely identify the corresponding level members in the underlying dataset by using either the values in a single column or the values in two or more columns (also known as a compound key) to identify each member of the level.



Note: AtScale does not automatically create compound keys for you by combining higher level key values with lower level key values.

- ▶ AtScale groups members by level key values instead of level name values.

Example

To illustrate these points, let's say you have a Date dimension in a model you're working on. On the model Canvas, you'll see the dimension.

Figure 1. The Date dimension

Date Dimension		
Date Month Hierarchy		
↳ Year	calendaryear	Int
↳ Quarter	YearQuarterKey	Int
↳ Month	YearMonthKey	Int
↳ Day	datekey	Long
Date Week Hierarchy		
↳ Year	calendaryear	Int
↳ Week	YearWeekKey	Int
↳ Day	datekey	Long

The dimension consists of two hierarchies: Date Month and Date Week. The key column used for each level appears in gray text below the level names, with the data types of the key columns displayed on the right.

WARNING: AtScale does NOT recommend mapping a level key or a dimensional attribute key to a floating point number data warehouse column. Doing so will prevent AtScale from applying certain performance optimizations. This is especially applicable to large result sets for Excel queries, but applies to all business intelligence tool queries.

To better understand how the hierarchies were modeled, we'll examine in detail the Date Month hierarchy. After we explain how the requirements were applied to this hierarchy, you'll also understand how they were applied to the Date Week hierarchy.

Click the context menu at the top of the Date dimension and select **Edit** to open the dimension Canvas. The dimension Canvas also displays the dataset that the dimension is based on. Here are its columns in descending alphabetical order, with calculated columns (highlighted in green) at the top.

Figure 2. The dataset for the Date dimension

Column Name	Data Type
# YearMonthKey	Int
# YearQuarterKey	Int
# YearWeekKey	Int
# calendarquarter	Int
# calendaryear	Int
# datekey	Long
# daynumberofmonth	Int
# daynumberofweek	Int
A englishdaynameofweek	String
A englishmonthname	String
A fulldatealternatekey	String
# monthnumberofyear	Int
# weeknumberofyear	Int

Notice that there are more columns here than appear in the view of the dimension that showed only the hierarchies, the levels, and the key for each level. In that view, there were five different columns, but in the dataset there are thirteen columns. That's because some of the columns are used as key columns, some as value columns, and some as secondary attributes.

Here are descriptions of the columns that are used in the Date Month hierarchy:

Table 1. The columns that are used in the Date Month hierarchy in the Date dimension

Name	Data Type	Description
YearMonthKey	Int	This is a calculated column. Its values concatenate years together with number identifiers for months. Example values:

		200501, 200502, 200503, etc.
YearQuarterKey	Int	This is a calculated column. Its values concatenate the years in calendaryear together with the quarters in calendarquarter. Examples values: 20051, 20052, 20053, etc.
calendarquarter	Int	Integers 1 through 4.
calendaryear	Int	Integer values in the format YYYY, such as 2005, 2006, 2007, etc.
datekey	Long	Long values in the format YYYYMMDD, such as 20050102, 20050103, 20050104, etc.
daynumberofweek	Int	Integers 1 through 7.
englishdaynameofweek	String	One string value per day of the week, such as Sunday, Monday, Tuesday, etc.
englishmonthname	String	One string value per month, such as January, February, March, etc.
fulldatealternatekey	String	Strings in the format YYYY-MM-DD, such as 2005-01-02, 2005-01-03, 2005-01-04, etc.

So that you have a better understanding of the data in each column, here is a table of four rows in each:

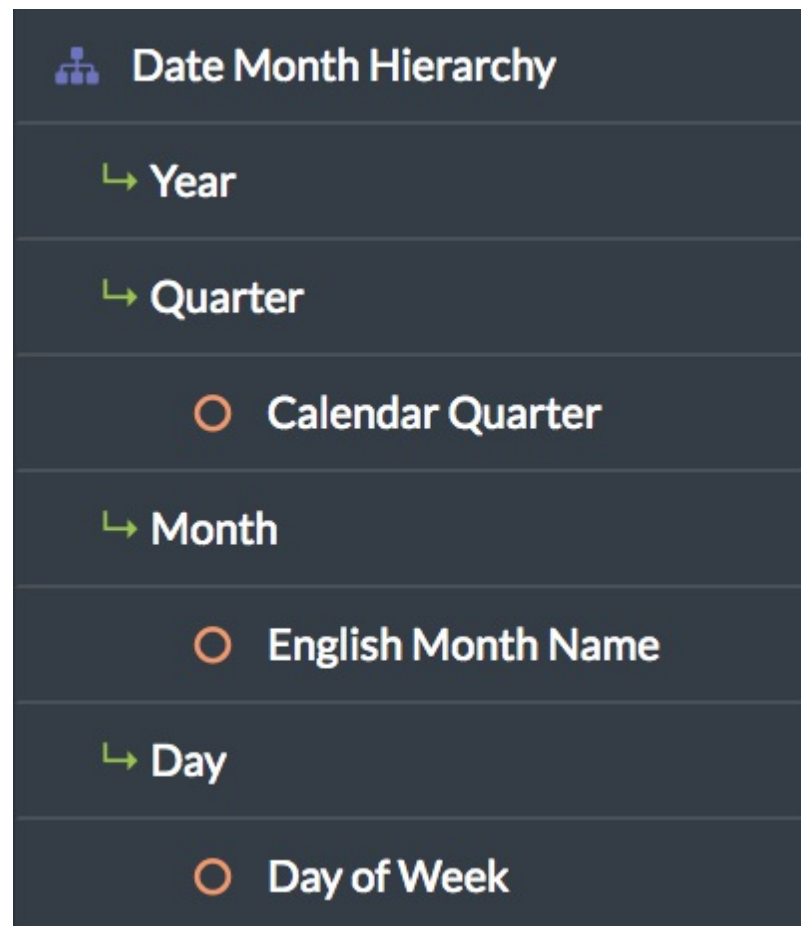
Table 2. Four rows of data in the columns of the Date Month hierarchy

calendarquarter	calendaryear	datekey	daynumberofweek	englishdaynameofweek	englishmonthname
1	2017	20170102	2	Monday	January

2	2017	20170404	3	Tuesday	April
3	2017	20170705	4	Wednesday	July
4	2017	20171005	5	Thursday	October

Here is another view of the Date Month hierarchy, the view that appears on the right side of the dimension Canvas. The items that are marked with circles are secondary attributes.

Figure 3. The hierarchies in the Date Month hierarchy of the Date dimension



When you add levels to a hierarchy, you can specify the key and value columns for each. You can also indicate whether every single value in a key column is unique to every row in the dataset.

This table lists the key and value columns used for each level and secondary attribute in the hierarchy.

Table 3. The Date Month hierarchy, showing the key columns and name (or value) columns for each level and secondary attribute

Type	Level or Secondary Attribute	Key	Unique?	Value
Level	Year	calendaryear	No	calendaryear

Level	Quarter	YearQuarterKey	No	YearQuarterKey
Secondary Attribute	Calendar Quarter	calendarquarter	N/A	calendarquarter
Level	Month	YearMonthKey	No	YearMonthKey
Secondary Attribute	English Month Name	englishmonthname	N/A	englishmonthname
Level	Day	datekey	Yes	fulldatealternatekey
Secondary Attribute	Day of Week	daynumberofweek	N/A	englishdaynameofweek

The reasons that AtScale defined the levels and secondary attributes with these columns are explained below.

Year

This is the highest level in the hierarchy. Aggregates defined on this level do not need to be rolled up for queries on higher levels. Therefore, using values of the format YYYY produces key values of sufficient uniqueness and do not need to be unique in each row of the table. The value column can be the same as the key column. No secondary attribute is necessary, as the values in the column can be used directly in reports without the reports losing clarity.

Quarter

The YearQuarterKey column is suitable for use as the Quarter key because it defines all quarter level members by combining year and quarter data in a single column. Similarly, it is valid to specify the calendaryear and calendarquarter columns together as a compound key.



Important: Suppose, however, that a column in the dataset (call it `year_month` and give it the data type String) did exist and had the same granularity as a compound key that consisted of calendaryear and calendarquarter. You might assume that you needed to specify only calendarquarter for the key column for the Quarter level, and that AtScale would automatically combine calendaryear, the key column for the higher Year level, with calendarquarter to implicitly create a compound key for the Quarter level. However, this assumption would be incorrect. AtScale does not automatically combine keys from higher levels with keys from lower levels to create compound keys for those lower levels. The key for Year is not automatically combined with the key specified for Quarter to implicitly create a compound key for Quarter.

Aggregates defined on this level can also satisfy queries on the Year level because queries can sum values for the quarters in queried years. This is in fact the purpose of arranging dimensional attributes in hierarchies: to specify which data it is possible to obtain from other, aggregated data. Suppose you ran a query to find out the sum of sales per quarter for a given year. The engine might subsequently create an aggregate based on the query, and the aggregate could group data by both year and month. So, if a later query asked for a sum of sales for a single year, say 2016, the query could use the existing aggregate, deriving the sum from the quarters in 2016.

Calendar Quarter

The values of `calendarquarter` are simply 1 through 4. In your reports, you can display these values to identify quarters, rather than the `YYYYQQ` values that are used in the level.

Month

The `YearMonth` column is suitable for use as the Month key because it defines all month level members by combining year and month data in a single column. Similarly, it is valid to specify the `calendaryear` and `monthnumberofyear` columns together as a compound key.

Aggregates defined on this level can also satisfy queries on the Year and Quarter levels.

Calendar Month

The values of `englishmonthname` are more readable in reports than numbers in the format `YYYYMM`.

Day

This is the lowest level of the hierarchy. Each member is identified by the combination of year, month, and day of month into values in the format `YYYYMMDD`. The Unique check box is selected because each value of the `datekey` column is unique within the dataset.

Day of Week

The values of `englishdaynameofweek` are more readable in reports than numbers in the format `YYYYMMDD`.