

# Using Custom Empty Members For Levels And Attributes

The Custom Empty Member functionality allows fact data with NULL or invalid foreign key values to be isolated and independently aggregated from fact records with valid foreign key values. Because fact records with invalid foreign keys are aggregated separately from records referencing valid dimension members, analysts can easily spot data integrity problems and further investigate them.

Use this feature to ensure that un-joinable values are included in query results and aggregated under a specially designated dimension member called the "Custom Empty Member". This feature is available when adding/editing [levels](#) and [secondary attributes](#).

## Custom Empty Members And BI Tools

To ensure the Custom Empty Member definition can be used with BI Tool features such as hierarchical drill-downs, top-n queries, dimension-only queries, and filtering, you must provide an empty record in the dimension's dataset.

To provide an empty record in the dataset, insert a record into the dimension table or create a query dataset with a UNION statement.



**Note:** The Custom Empty Member values you use must be different than the values used by existing members in the dimension table. Additionally, the chosen key value must not be used in the fact table.

The Custom Empty Member types for Keys and Name properties must match the dimension column types. For example, to configure the Sales Reason dimension to work with un-joinable fact records, do the following:

1. Insert a place-holder dimension record for unknown dimension values.

If you cannot insert data into the dimension table you can include a record at query time by configuring the dimension's dataset to use a `SELECT` statement with a `UNION` clause. To do this:

1. Open the Order Dimension Canvas and edit the SalesReason dataset.
2. Change the **Source Type** from **Table** to **Query**, and enter the following SQL:

```
select salesreasonkey, salesreasonname, salesreasongroup from schema.dimsalesreason union select 0, 'UNKNOWN REASON', 'UNKNOWN GROUP'
```

2. Save the dimension's dataset.
3. Open the **Edit Dimension** panel for the Order dimension. In the **Hierarchies** section, open the Sales Reason hierarchy level for editing, and check the **Custom Empty Member** checkbox. The system displays inputs for each key and name field use to define the level or secondary attribute.

4. Enter empty member values for each Key Field and the Name Field and save the level. Note the following:
  - ▲ The values must match the values used in your dataset QDS or the unknown dimension record.
  - ▲ The values must be convertible to the data types of the corresponding columns. Given the compound key columns of `salesreasonkey` (Long) and `salesreasongroup` (String), enter values of `0` and `UNKNOWN GROUP`. Note that you must enter empty member values for the keys. Given the level Name column of `salesreasonname` (String) enter a value of `UNKNOWN REASON`.
  - ▲ If your Level or Secondary Attribute specifies a custom sort column, then you must also specify a Customer Empty Member value for the custom sort column. Note the following:
    - ▲ If a value for the custom sort column is not entered then the project cannot be deployed.
    - ▲ AtScale regularly checks for missing values in the Sort Name field, and displays warnings about this.
5. Repeat step 4 for all other levels for which you require empty member support.
6. Deploy the project.

The unknown records will now participate in Top-N measure queries as well as Hierarchical drill-downs.



**Note:** If the **Custom Empty Member** option is enabled for a specific level or attribute, there must be a chain of Custom Empty Member configurations between that attribute and the level used in a relationship. For example, if you have a three-level hierarchy that joins to the fact table at the lowest level, and you want Custom Empty Member behavior at the root level, then it is required that you configure Custom Empty Members at the root level, the lowest level, and the intermediate level. Similarly, for secondary attributes, you must configure a Custom Empty Member for the containing level that joins to the fact table.

## Nested Dimension Custom Empty Member Behavior

When grouping a fact dataset's measures by a nested dimension hierarchy that has an applicable Custom Empty Member configuration, AtScale uses outer joins to include all records in the intermediate tables, as well as the selected fact dataset. AtScale does this to ensure that records orphaned due to any invalid key along the join path are included in the Custom Empty Member grouping. This behavior ensures that the nested dimension subtotals sum to the same value as the Grand Total queried directly against the fact dataset.

If you have a custom business case that requires the exclusion of some orphaned records, then you can disable this behavior. By doing so, the system uses the Custom Empty Member configuration of each hierarchy to define the join to the non-arrow end of the relationship.

To disable the default custom empty member behavior for nested dimensions, set the `query.planning.independentCustomEmptyMembers` engine setting to `false`, then redeploy your catalog. For instructions on changing engine settings, see [Changing Engine Settings](#).

Here is an example how disabling the default behavior works. Consider the following system:

- ▲ The `query.planning.independentCustomEmptyMembers` engine setting is set to false.
- ▲ There are a Reseller Fact table, a Reseller Dimension, and a nested Reseller Geography Dimension.
- ▲ The Reseller Dimension Hierarchy has no CEM configuration, and the nested Reseller Geography Dimension Hierarchy has a CEM configuration.

Selecting ResellerSales grouped by `[ResellerGeoDim].[ResellerGeoHierarchy].[Country]` will result in a Left Outer Join between the Reseller table and the ResellerGeography table, and an Inner Join between the ResellerFact table and the Reseller table. The resulting query will omit ResellerFact records that do not have a valid Reseller foreign key, but will include ResellerFact records that have a valid Reseller foreign key and an invalid `Reseller.ResellerGeographyID` key. Note that in this configuration, other queries against ResellerFact will include the records without valid Reseller keys, resulting in Totals that may not equal the sum of the Grouped Reseller Country results.

## How Null Values Affect Aggregates

Because most data warehouses do not include rows with `NULL` join key values in the result set, aggregate tables will be incomplete if your fact or dimension tables contain `NULL` values in the columns used as relationship keys. To ensure that queries against your aggregate tables match results against the non-aggregates fact tables you **must** do one of the following:

- ▲ Augment your ETL scripts to ensure that `NULL` values are not written to key columns used in relationships.
- ▲ Configure AtScale's Custom Empty Member feature for every dimension that could have `NULL` key values in either the dimension table or in a related fact table.