Module 4: Building Dimensions Using the Dimension Editor

Overview
- Understanding Dimension Basics
- Shared vs. Private Dimensions
- Working with Standard Dimensions
- Basic Level Properties
- Working with Parent-Child Dimensions

Understanding Dimension Basics
- Enabling Various Views
- Understanding Levels and Members
- Describing Familial Relationships
- Reviewing Analysis Services Limits

Enabling Various Views
- Analysis Server
- Finance
  - Profit by Division
  - Profit by Country
  - Profit by Month
  - Profit by Actual/Budget
- Operations
  - Revenue by Product
  - Revenue by Region
  - Revenue by Sales Rep
  - Revenue by Quarter
  - Volume by Plant
  - Volume by Shift
  - Volume by Product
  - Volume by Day
- Marketing
  - Revenue by Customer
  - Revenue by Industry
  - Revenue by Channel
  - Revenue by Week
- Marketing
  - Revenue by Customer
  - Revenue by Industry
  - Revenue by Channel
  - Revenue by Week

Understanding Levels and Members
- Product Dimension
- Four Levels: All, Category, Sub-Category, Product
- Category Members: Bread, Dairy, Meat

Describing Familial Relationships
- Region
  - Central
    - IL
    - MO
  - East
    - NY
    - MA
- Ancestors of IL and MO
- Descendants of Region
- Cousins
- Parents
- Siblings
- Children
Reviewing Analysis Services Limits

<table>
<thead>
<tr>
<th>Items</th>
<th>Limits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimensions per database</td>
<td>65,535</td>
</tr>
<tr>
<td>Levels per database</td>
<td>65,536</td>
</tr>
<tr>
<td>Dimensions per cube</td>
<td>128</td>
</tr>
<tr>
<td>Levels per cube</td>
<td>256</td>
</tr>
<tr>
<td>Levels per dimension</td>
<td>64</td>
</tr>
<tr>
<td>Members per parent</td>
<td>64,000</td>
</tr>
<tr>
<td>Length of dimension name</td>
<td>24 characters</td>
</tr>
</tbody>
</table>

◆ Shared vs. Private Dimensions

- Working with Shared Dimensions
- Working with Private Dimensions

Working with Shared Dimensions

- Created Once and Shared by One or More Cubes in a Database
- Cannot Be Changed to Private
- Maintained in Dimension Editor
- Administered in One Place
- Cause All Cubes Using that Dimension to be Unavailable for Querying After Rebuilding Structure
- Identified by a Sharing Hand Icon:

Working with Private Dimensions

- Created and Used within Single Cube
- Maintained in Cube Editor, Not Dimension Editor
- Cannot Be Changed to Shared
- Rebuilt Automatically with Cube Process
- Identified by Dimension Icon: 

Working with Standard Dimensions

- Each Level Corresponds to a Dimension Table Column
- All Members at a Given Level Have the Same Number of Ancestors

◆ Basic Level Properties

- Assigning Member Keys and Names
- Identifying Uniqueness of Members
- Creating Members from Expressions
- Working with Ragged Dimensions
- Understanding Snowflake Dimensions
- Defining the All Level
- Specifying a Default Member
Assigning Member Keys and Names

- **Defining the Member Key Column**
  - Determines the members included in a level
  - Usually comes from a single dimension table column

- **Defining the Member Name Column**
  - Provides names for members at a level
  - Can be different from the Member Key Column

- **Sorting Members in a Level**
  - Order by key
  - Order by name
  - Order by member property

Identifying Uniqueness of Members

- **Member Keys Unique**
  - Works with the Member Key Column
  - Is set as a dimension or level property
  - Affects cube processing performance with True setting
  - Cannot be set within a parent-child dimension

- **Member Names Unique**
  - Works with the Member Name Column
  - Is set as a dimension or level property
  - Affects member naming in MDX with True setting

Creating Members from Expressions

- **Add Flexibility When Defining Levels**
- Are Created from One or More Columns in a Single Table
- Are Defined in the Member Key Column and Member Name Column in the Dimension Editor
- Act as RDBMS Pass-Through Functions
- Must be Valid RDBMS Syntax

Working with Ragged Dimensions

- Variable Depth in Branches
- Level Property Hide Member If

Understanding Snowflake Dimensions

- Dimension Tables
- Fact Table

Defining the All Level

- Summarizes All Data at Top Level of Dimension
- Is Included by Default
- Is Named All DimensionName by Default
  - For example, All Product
- Can Be Turned Off within the Dimension Editor
- Cannot Be Defined by the Member Key Column or the Member Name Column
- Can Be Renamed Using the All Caption Property
Specifying a Default Member

Lab A: Creating a Standard Dimension

Lab B: Creating a Snowflake Dimension

Working with Parent-Child Dimensions

Overview of Parent-Child Dimensions
- Are Based on a Two Column Dimension Table
- Contain Levels Created by Parent-Child Relationships
- Contain Unbalanced Levels
- Are Created with the Dimension Wizard
- Can Slow Queries that Reference Them

Structure of a Parent-Child Dimension

<table>
<thead>
<tr>
<th>Employee</th>
<th>Manager</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith</td>
<td>&lt;none&gt;</td>
</tr>
<tr>
<td>Jones</td>
<td>Smith</td>
</tr>
<tr>
<td>White</td>
<td>Smith</td>
</tr>
<tr>
<td>Block</td>
<td>Jones</td>
</tr>
<tr>
<td>Hart</td>
<td>Jones</td>
</tr>
<tr>
<td>Knight</td>
<td>Jones</td>
</tr>
<tr>
<td>Fox</td>
<td>Hart</td>
</tr>
<tr>
<td>Hunt</td>
<td>Hart</td>
</tr>
<tr>
<td>Smart</td>
<td>Hunt</td>
</tr>
</tbody>
</table>
Members with Data

- In Standard Dimensions, Only Leaf Members Can Correspond to Fact Table Data
- In Parent-Child Dimensions, Leaf and Upper Level Members Correspond to Fact Table Data
- The Members with Data Property Has Three Possible Settings:
  - Leaf Members Only
  - Non-leaf Data Hidden
  - Non-leaf Data Visible

Parent-Child Level Management

Skipped Levels Column

- Creates Ragged Hierarchies within Parent-Child Dimensions
- Uses a Column in the Dimension Table that Contains the Number of Levels to Skip
- Is Accessed from the Advanced Tab of the Properties Pane

Lab C: Creating a Parent-Child Dimension

Review

- Understanding Dimension Basics
- Shared vs. Private Dimensions
- Working with Standard Dimensions
- Basic Level Properties
- Working with Parent-Child Dimensions