sqlbi

sqlbi.com

Analysis Services Best Practices

presented by
Marco Russo
marco@sqlbi.com

Who am I

- BI Expert and Consultant
- Problem Solving
- Complex Project Assistance
- DataWarehouse Assesments and Development
- Courses, Trainings and Workshops
- Microsoft Business Intelligence Partner
- Book Writer


Latest conferences

- PASS Europe 2009 – Neuss – Germany
- PASS 2009 – Seattle – USA
- SQL Conference 2010 – Milan – Italy
- Teched 2010 – New Orleans – USA
- 24 Hours of PASS 2010 – Online
- PASS 2010 – Seattle – USA

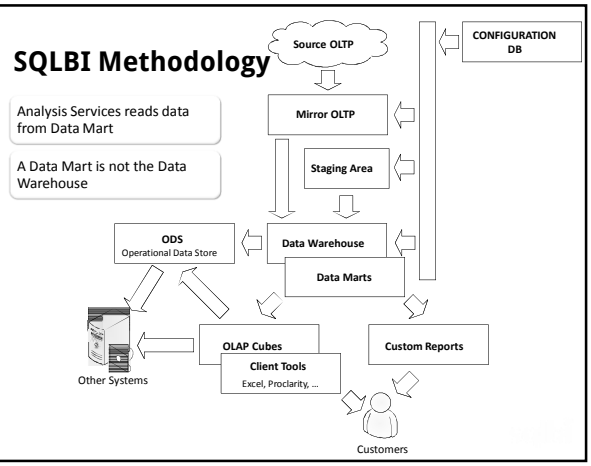
Agenda

- Relational Schema
- Decoupling Layer
- Dimensional Patterns
- Slowly Changing Dimensions
- Junk Dimensions
- Parent-Child Hierarchies
- Role Dimensions
- Drill-through
- Calculation Dimensions

DATA SOURCE (RELATIONAL MODELING)

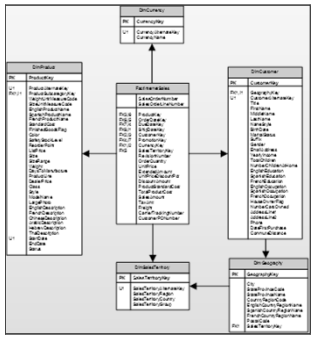
Relational Schema

SQLBI Methodology



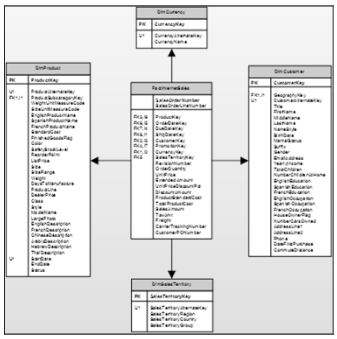
Relational Schema

SNOWFLAKE SCHEMA



Relational Schema

STAR SCHEMA



Relational Schema

STAR VS. SNOWFLAKE SCHEMA

- Options for dimensions from snowflake schema:
  - Transform into a star schema by using views
  - Transform into a star schema by using DWV queries
  - Join tables in SSAS dimensions
  - Referenced Dimension
- Ideal solution
  - Use SQL views to generate a star schema
  - The star schema eliminates ambiguity

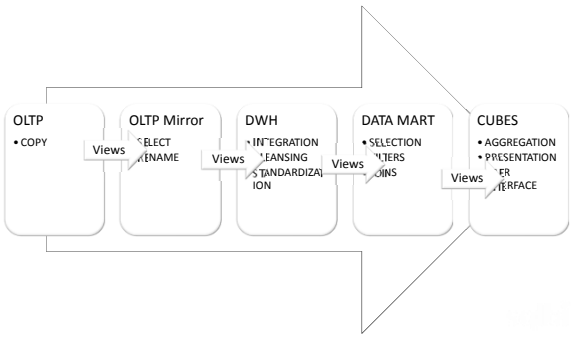
DATA SOURCE (RELATIONAL MODELING)

Decoupling Layer

sqlbi.com

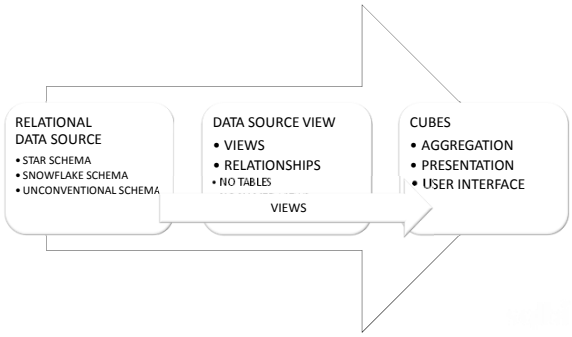
Data Source Decoupling

USE VIEWS TO DECOUPLE DIFFERENT LAYERS OF A BI SOLUTION



Data Source Decoupling

DECOUPLE SSAS MODEL – RDBMS STRUCTURE



Data Source Decoupling

DEFINE VIEWS FOR SSAS CUBES

Model

- One view for each dimension
- Namespace = cube name (Common for shared dimensions, if necessary)
- Convert a snowflake in a star schema for SSAS

Columns

- Number of view's columns = number of attributes of dimension
- Column name = attribute name
  - Use a naming convention for attribute keys
- Simple calculations or string operations in views
  - Avoid calculated column in DSV
- Default values when necessary
  - Eliminates NULL (i.e. in case of a LEFT JOIN)

Data Source Decoupling

WHY TO USE VIEWS?

Decoupling interface	Stored in DB	Textual
Easy to change	Database objects	Can be optimized

SURROGATE KEY, ATTRIBUTE KEY
GROUPING, BANDING

Dimensional Patterns

sqlbi.com

Surrogate Keys

WHY YOU SHOULDN'T EXPOSE SURROGATE KEYS

Benefits

- Independency from application keys
- SCD modeling
- Star schema optimization

Drawbacks with Analysis Services

- MDX query with surrogate keys
  - Don't have a semantic value
  - Are invalidated if the cube is reprocessed (i.e. nightly one-shot processing)
- Impact on custom security and reporting

Surrogate Keys

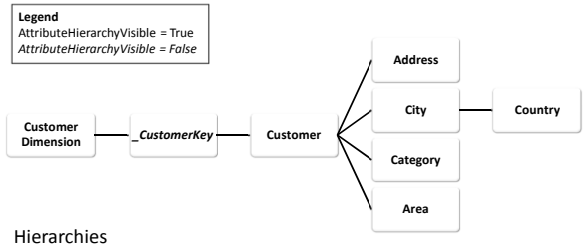
PATTERN FOR NOT EXPOSING SURROGATE KEYS IN SSAS

Dimensione Customer

- _CustomerKey
  - KeyColumns: ID_Customer (surrogate key)
  - NameColumn: (none)
  - AttributeHierarchyVisible: False
  - AttributeRelationship
    - Attribute: Customer
    - Cardinality: One
- Customer
  - KeyColumns: COD_Customer (business key)
  - NameColumn: CustomerName
  - AttributeRelationship
    - SCD Type I – all other attributes are related to this one
    - SCD Type II – evaluate on a case-by-case basis (impact on KeyColumns)

Surrogate Keys

PATTERN ATTRIBUTE HIERARCHY FOR SCD TYPE I

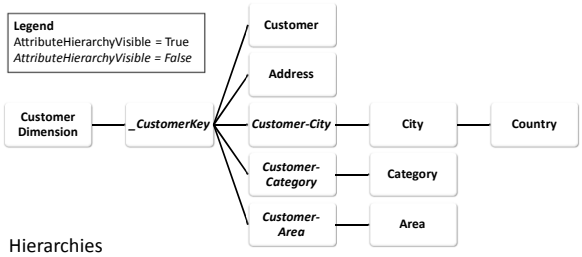


Hierarchies

- Country / City / Customer
- Category / Customer
- Area / Customer

Surrogate Keys

PATTERN ATTRIBUTE HIERARCHY FOR SCD TYPE II



Hierarchies

- Country / City / Customer (*Customer-City*)
- Category / Customer (*Customer-Category*)
- Area / Customer (*Customer-Area*)

Attribute Keys

PATTERN FOR CREATING NATURAL HIERARCHIES FROM CUSTOM HIERARCHIES

Use unique key for attribute

- i.e. Subcategory code if it is unique
- Don't create one in ETL if it doesn't exist (use composite key instead)

If it doesn't exist, use attribute with composite key

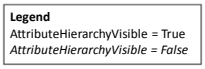
- Set of keys corresponding to attribute hierarchy
- Set AttributeHierarchyVisible = False

Use decoupling views to generate columns for appropriate descriptions

- Natural hierarchy: you may reference the parent's name
- Browsable attribute: use a simple description
  - i.e. «February 2010» hierarchy - «February» attribute

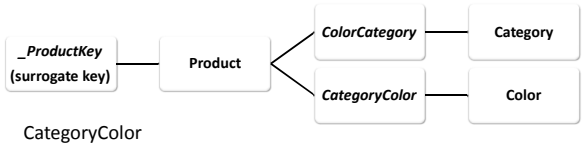
Attribute Keys

PATTERN FOR PRODUCT DIMENSION



Hierarchies

- Color / Category (*CategoryColor*) / Product
- Category / Color (*ColorCategory*) / Product



CategoryColor

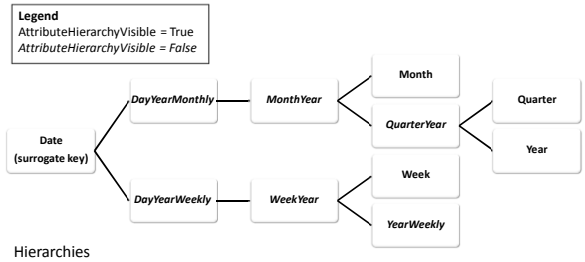
- KeyColumns: COD_Category, COD_Color
- NameColumn: Category

ColorCategory

- KeyColumns:COD_Color, COD_Category
- NameColumn: Color

Attribute Keys

PATTERN FOR DATE DIMENSION



Hierarchies

- Year / Quarter (*QuarterYear*) / Month (*MonthYear*) / Day (*DayYearMonthly*)
- Year / Month (*MonthYear*) / Day (*DayYearMonthly*)
- Year (*YearWeekly*) / Week (*WeekYear*) / Day (*DayYearWeekly*)

Grouping

ATTRIBUTE GROUPING – AUTOMATIC VS. MANUAL

Automatic grouping

- DiscretizationMethod
  - EqualAreas (same number of elements)
  - Clusters (Data Mining algorithm)
  - Automatic
- DiscretizationBucketCount
  - Number of groups

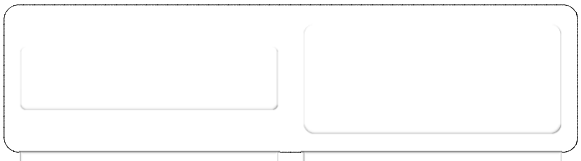
Manual grouping

- Business logic in view for dimension on Data Mart

```
CASE WHEN Weight IS NULL
          OR Weight<0 THEN 'N/A'
     WHEN Weight<10 THEN '0-10Kg'
     WHEN Weight<20 THEN '10-20Kg'
     ELSE '20Kg or more'
END
```

Banding

CATEGORIZE A MEASURE ACCORDING TO RANGES OF VALUES



Direct relationship

- Range granularity is the same as dimension granularity
- Impact of range change
  - Fact table ETL
  - Dimension ETL
  - Reprocess Dimension
  - Reprocess Cube

Indirect relationship

- Discretization and constant granularity
- Impact of range change
  - Dimension ETL
  - Reprocess Dimension

SSAS DESIGN FOR SCD TYPE I AND TYPE II

Slowly Changing Dimensions in SSAS

sqlbi.com

Type I SCD

ISSUES WITH PROCESS UPDATE

Process Update

- Doesn't detect duplicate key errors
  - Process Full would fail according with error handling settings
- Flexible aggregations are regenerated

Existing MDX Queries

- Unique Name may change
- MDXMissingMemberMode – Ignore (Default)
  - Missing elements are ignored
  - Rows/columns disappear without a warning
- MDXMissingMemberMode – Error
  - Need to change the query
  - Some clients require to rebuild the query from scratch

Type II SCD

SSAS MODELING

Process Dimension

- Process Add instead of Process Update
- Attribute relationship: always Rigid

SSAS Dimension Modeling

- There is no specific dimension type for SCD
- Surrogate key → CustomerKey
  - CustomerSCD attribute – (Primary key attribute)
- Business key → CustomerAlternateKey
  - Customer attribute

Type II SCD

ATTRIBUTE RELATIONSHIP AND HIERARCHIES

"Standard" Modeling

- Only «visible» attributes
- CustomerSCD should be invisible
  - Otherwise may be confusing for end user
- Country-City-Customer hierarchy is not natural

"Ideal" Modeling

- Invisible attribute "Customer in City"
- Composite key City + CustomerAlternateKey
- Used only in hierarchy Country-City-Customer at Customer level

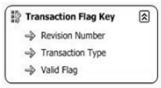
CONSOLIDATE INDEPENDENT ATTRIBUTES

Junk dimension

sqlbi.com

Junk Dimension

MODELING



Junk dimension in relational modeling

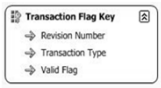
- Ideal solution for performance and consumed space
- New attribute requires reprocess of fact table
  - Unless a valid default exists for existing data

Junk dimension only on cube

- Create dimension based on a view
  - Use CROSS JOIN
  - Creates also combinations of values that are never used
  - A new attribute always require cube reprocess

Junk Dimension

PERFORMANCE AND SEMANTIC



Reduce aggregation complexity

- Fewer dimensions
- Same number of attributes, but optimization on AttributeKey more efficient
- Better performance in both query and process

Usability

- Improves if dimension name is intuitive
- Use more than one junk dimension if a single name is not good
- Set Attribute Key hierarchy to invisibile
- Carefully choose names
  - Otherwise you might invalidate existing queries by changing them later

LIMITATIONS OF STANDARD PARENT-CHILD HIERARCHIES
MODELING MULTIPLE HIERARCHIES

Parent-child Hierarchies

sqlbi.com

Parent-Child Hierarchies

LIMITATIONS

Functional limits

- Only one parent-child hierarchy per dimension
- Must use the dimension's Attribute Key
- Unique name for a member includes surrogate keys

Performance issues

- Increase processing time
- MDX calculation is more complex (especially SCOPE, autoexists e CurrentMember)
- Slower query response (Good up to some thousand members)

Alternative

- Parent-child naturalizer in BIDS Helper

Parent-Child Hierarchies

STANDARD PARENT-CHILD HIERARCHIES IN ANALYSIS SERVICES

Pros

- Internal optimization is good enough (if not too many members)
- Easy to create and maintain for SSAS developer

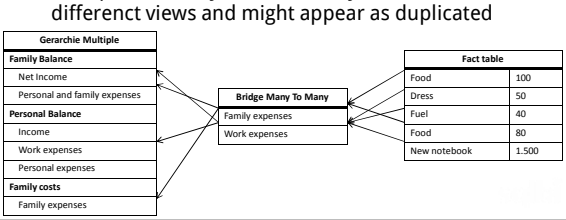
Cons

- Rigid structure
- Only one parent-child hierarchy per dimension
- User cannot add custom parent-child hierarchies
- Defining a separate dimension for each parent-child hierarchy (made with the same leaf nodes) increases the number of aggregations

Parent-Child Hierarchies

STANDARD PARENT-CHILD VS MULTIPLE HIERARCHIES

- A multiple hierarchy IS NOT a parent-child with just some more nodes
  - Parent-child data is partitioned
  - Multiple hierarchy data are always the same but with differenct views and might appear as duplicated



Parent-Child Hierarchies

MULTIPLE HIERARCHIES

Pros

- Flexible structure (data-driven approach)
- A single dimension defines many hierarchies by using the same data
- User can define new hierarchies without changing the cube structure
- A single dimension to aggregate is better
- Easy to use? It depends…

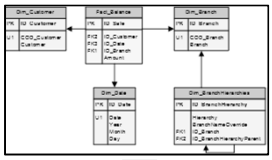
Cons

- Optimization is in charge of the SSAS developer
- They are complex to create and maintain

Parent-Child Hierarchies

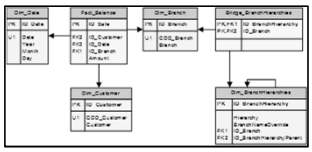
RELATIONAL MODEL FOR MULTIPLE HIERARCHIES

Standard Parent-Child
The relationship with Fact_Balance is direct, each transaction has a related account



Multiple Hierarchies
The relationship with Fact_Balance is made through a many-to-many relationship by using a bridge table

The Branch dimension loses the hierarchy and is flat



PROS & CONS, POSSIBLE ALTERNATIVES

Role Dimensions

Role Dimensions

PROS AND CONS

Pros

- Only one physical dimension in Analysis Services
- A change in a dimension applies to all the “roles”
- A single dimension process is required

Cons

- A single “role” cannot be modified
- Attribute and hierarchy names cannot be changed
- User interface in some clients makes it difficult to understand to which role each hierarchy belongs to

Role Dimensions

ALTERNATIVES

Duplicate dimension in SSAS

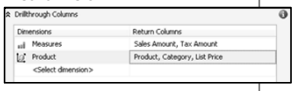
- Create a view for each dimension
- The view can filter rows/columns
- Rename fields in a meaningful name (for the role)
- Name of hierarchies can be role-based
- Keep an alignment between view and role

COLUMN ORDER
CALCULATED MEMBERS
MEASURES OR DIMENSIONS
MOLAP VS ROLAP

Drill-through

Drillthrough

COLUMNS ORDER



Change XML

- BIDS editor doesn't support column order mixing several dimensions
- After a change in MDX the graphical editor cannot be used anymore (otherwise you would lose custom order)

```
<Actions>
<Action xsi:type="DrillThroughAction">
  <ID>Drillthrough Action</ID>
  …
  <Columns>
    <Column xsi:type="MeasureBinding">
      <MeasureID>Sales Amount</MeasureID>
    </Column>
    <Column xsi:type="MeasureBinding">
      <MeasureID>Tax Amount</MeasureID>
    </Column>
    <Column xsi:type="CubeAttributeBinding">
      <CubeID>Adv DM 1</CubeID>
      <CubeDimensionID>Product</CubeDimensionID>
      <AttributeID>Products</AttributeID>
      <Type>All</Type>
    </Column>
    <Column xsi:type="CubeAttributeBinding">
      <CubeID>Adv DM 1</CubeID>
      <CubeDimensionID>Product</CubeDimensionID>
      <AttributeID>Category</AttributeID>
      <Type>All</Type>
    </Column>
    <Column xsi:type="CubeAttributeBinding">
      <CubeID>Adv DM 1</CubeID>
      <CubeDimensionID>Product</CubeDimensionID>
      <AttributeID>List Price</AttributeID>
      <Type>All</Type>
    </Column>
  </Columns>
</Action>
</Actions>
```

Drillthrough

ORDINAMENTO COLONNE

Change XML

- Graphical editor may show an incorrect order
- Always group by measures or dimensions

```
<Actions>
<Action xsi:type="DrillThroughAction">
  <ID>Drillthrough Action</ID>
  …
  <Columns>
    <Column xsi:type="MeasureBinding">
      <MeasureID>Sales Amount</MeasureID>
    </Column>
    <Column xsi:type="CubeAttributeBinding">
      <CubeID>Adv DM 1</CubeID>
      <CubeDimensionID>Product</CubeDimensionID>
      <AttributeID>Products</AttributeID>
      <Type>All</Type>
    </Column>
    <Column xsi:type="CubeAttributeBinding">
      <CubeID>Adv DM 1</CubeID>
      <CubeDimensionID>Product</CubeDimensionID>
      <AttributeID>List Price</AttributeID>
      <Type>All</Type>
    </Column>
    <Column xsi:type="MeasureBinding">
      <MeasureID>Tax Amount</MeasureID>
    </Column>
    <Column xsi:type="CubeAttributeBinding">
      <CubeID>Adv DM 1</CubeID>
      <CubeDimensionID>Product</CubeDimensionID>
      <AttributeID>Category</AttributeID>
      <Type>All</Type>
    </Column>
</Columns>
</Action>
</Actions>
```

Drillthrough

NOT SUPPORTED ON CALCULATED MEMBERS

- Calculated measures
  - Create regular measures and use SCOPE assignments
  - Cost: MOLAP space for measure storage
- Calculated members on dimensions
  - Works with utility dimensions (i.e. DateTool)
  - Consider semantic meaning (i.e. YTD)
- Use Action – Rowset
  - You can specify a DRILLTHROUGH operation
  - Query generated server-side
    - Cannot be controlled by the client – i.e. MAXROWS

Drillthrough

OPTIONS FOR NUMERICAL ATTRIBUTES (I.E. DOCUMENT NUMBER)

Transaction Detail Dimension

- Supports supplementary information (also textual)
- Long process time (requires SELECT DISTINCT)
- Large dimension: consider to hide it

ROLAP Dimension

- Reduced processing time
- Heavy load at query time (full scan of fact table with a SQL query)

Measure on a separate fact table

- Only for numeric values
- Aggregation Function = None (only Enterprise)
- Same fact table if only one integer value is required
- Processing time affordable (doesn't require DISTINCT)

Drillthrough

MOLAP VS. ROLAP

MOLAP

- Requires SELECT DISTINCT for each dimension
- Longer processing time if a dimension is added just for drillthrough attributes
- Fast query time, both on measures and on large dimensions

ROLAP

- Process time is immediate
- Slow queries (pressure on relational engine, SQL queries)
- Almost never a good choice

TIME INTELLIGENCE WIZARD LIMITS
TECHNIQUES AND APPLICATION IN DATETOOL

Calculation Dimensions

sqlbi.com

Calculation Dimension

BASE CONCEPTS

Goal: apply the same calculation to different measures

- Year-to-Date
- Year-over-Year
- Difference over Year

Implementation

- "Virtual" space in the multidimensional model
- Each member in each attribute defines a calculation
- MDX Script contains the calculation formulas

Time Intelligence Wizard

WIZARD LIMITS IN BI DEVELOPER STUDIO

Attribute Overwrite

- Create attributes on existing hierarchy
  - http://tinyurl.com/attoverwrite
  - http://tinyurl.com/chrisattoverwrite

Wizard generated formulas have limits on multiple hierarchies

- i.e. Year-Month and Year-Week

Use calculated membersi

- Drillthrough cannot be used
- Security cannot be applied
- Issues with clients like Excel 2007

Calculation Dimension

DESIGN PATTERN

SQL Views

- View with member for each attribute
- CROSSJOIN between attributes for a dimension

Attributes properties

- Remove All Member (IsAggregatable = False)

Dimensions relationships

- Add dimension to the cube
- No relationship with any measure group

MDX Script

- Use SCOPE to assign the MDX formula to the corresponding member in the attribute

Conclusions

- Leverage on data modeling
  - Relational
  - Multidimensional
- Use decoupling views
- Avod MDX if possible
  - Calculate values inside ETL if you can
  - Simplify mainenance by splitting complex calculations in an higher number of simple calculation
- Hide "dangerous" attributes to the end-user
  - i.e. dimensions that supports drillthrough

Questions and Answers

sqlbi.com

Links

- SQLBI Website
  www.sqlbi.com
- PowerPivot Workshop
  www.powerpivotworkshop.com
- Marco Russo blog
  www.sqlblog.com/blogs/marco_russo
- Alberto Ferrari blog
  www.sqlblog.com/blogs/alberto_ferrari

For any question contact us at
info@sqlbi.com


Thank you!
sqlbi.com