Are Your Stars Aligned?
Better Data Models for Business Intelligence

Christina Rouse, Chief Architect
Incisive Analytics
chris.rouse@incisiveanalytics.com

April 11, 2013
Data Modelers

- How many data modelers does it take to create a database?
Need: A Two Way Alignment

The business intelligence star schema data model should:

• Be aligned to the business needs and user questions and not require any additional “desktop calculations”

• Be reconciled or aligned to itself and its sister star schemas
What is Business Intelligence?
Isn’t it the same as Data Warehousing?

**Business Intelligence** (BI) is the use of mathematical facts to improve business decisions. These facts typically focus on company performance across time and throughout the organization. BI is the use of a data warehouse.

**Data Warehousing** (DW) is the methodology and technology used to capture data from the company’s operational systems and then present the data in a meaningful way.

Business
'biz-n&s, -n&z
A usually commercial or mercantile activity engaged in as a means of livelihood

Intelligence
in-'te-l&-j&n(t)s
The ability to apply knowledge to manipulate one’s environment or to think abstractly as measured by objective criteria (as tests)

**DW is the technology. BI is the use of the technology to make better decisions that improve company performance.**
BI Visuals
Can you show some examples of BI?
A Two Stack Model

1. Data Warehouse – Physical star schema model
2. Business Intelligence – Semantic layer model
   • Hierarchy levels
   • Counters
   • Sliding Time Helpers
Star Schema Fundamentals

Join multiple stars for horizontal analysis

- Loss ratio
- Agent performance
- Actual to quota
- Quote win rate

Enable users to drop ‘n drag

- Ad hoc reports
- Cube drilling
Inner Joins Only

• Join fact to dimension only
• Join as inner join
• Avoid dimension to dimension join whenever possible
• Creates a consistent, controlled environment with no runaway queries
Why Stars Work
A Single Star Query

SELECT
STAR1_DIMENSION_1.COLUMN_1
STAR1_DIMENSION_1.COLUMN_N
STAR1_DIMENSION_2.COLUMN_1
STAR1_DIMENSION_2.COLUMN_N
STAR1_DIMENSION_N.COLUMN_1
STAR1_DIMENSION_N.COLUMN_N
STAR1_FACT.COLUMN_1
STAR1_FACT.COLUMN_N

FROM
STAR1_DIMENSION_1
STAR1_DIMENSION_2
STAR1_DIMENSION_N
STAR1_FACT

WHERE
STAR1_DIMENSION_1.KEY_1 = STAR1_FACT.KEY1_COLUMN
STAR1_DIMENSION_2.KEY_2 = STAR1_FACT.KEY2_COLUMN
STAR1_DIMENSION_3.KEY_3 = STAR1_FACT.KEY3_COLUMN
STAR1_DIMENSION_N.KEY_N = STAR1_FACT.KEYN_COLUMN

User Drop and Drag
Users may drop and drag any data elements onto a report.

Semantic Layer
All joins are INNER JOINs and managed by the semantic layer.

Tables used are controlled by the semantic layer and updated in only one location.

User Filter
The user enters filter criteria like Year and Term or Major.
Why Stars Work

A Cross Star Query

```sql
SELECT
    STAR1_DIMENSION1 COLUMN1,
    STAR1_DIMENSION1 COLUMNN,
    STAR1 DIMENSION2 COLUMN1,
    STAR1 DIMENSION2 COLUMNN,
    STAR1 DIMENSIONN COLUMN1,
    STAR1_DIMENSIONN COLUMNN,
    STAR1_FACT COLUMN1,
    STAR1_FACT COLUMNN,
    STAR2_DIMENSION1 COLUMN1,
    STAR2_DIMENSION1 COLUMNN,
    STAR2 DIMENSION2 COLUMN1,
    STAR2_DIMENSION2 COLUMNN,
    STAR2 DIMENSIONN COLUMN1,
    STAR2_DIMENSIONN COLUMNN,
    STAR2_FACT COLUMN1,
    STAR2_FACT COLUMNN,
FROM
    STAR1_DIMENSION1,
    STAR1_DIMENSION2,
    STAR1_DIMENSIONN,
    STAR1_FACT,
    STAR2_DIMENSION1,
    STAR2_DIMENSION2,
    STAR2_DIMENSIONN,
    STAR2_FACT,
WHERE
    STAR1_DIMENSION1 KEY1 = STAR1_FACT KEY1 COLUMN1,
    STAR1_DIMENSION1 KEY1 = STAR1_FACT KEY1 COLUMNN,
    STAR1_DIMENSION2 KEY2 = STAR1_FACT KEY2 COLUMN1,
    STAR1_DIMENSION2 KEY2 = STAR1_FACT KEY2 COLUMNN,
    STAR1_DIMENSIONN KEYN = STAR1_FACT KEYN COLUMN1,
    STAR1_DIMENSIONN KEYN = STAR1_FACT KEYN COLUMNN,
    STAR2_DIMENSION1 KEY1 = STAR2_FACT KEY1 COLUMN1,
    STAR2_DIMENSION1 KEY1 = STAR2_FACT KEY1 COLUMNN,
    STAR2_DIMENSION2 KEY2 = STAR2_FACT KEY2 COLUMN1,
    STAR2_DIMENSION2 KEY2 = STAR2_FACT KEY2 COLUMNN,
    STAR2_DIMENSIONN KEYN = STAR2_FACT KEYN COLUMN1,
    STAR2_DIMENSIONN KEYN = STAR2_FACT KEYN COLUMNN,
    Cross Star
    All joins fully qualified by TABLENAME.COLUMNSNAME
    Other conditions
    Fully qualified by TABLENAME.COLUMNSNAME
```
Star Schema Quiz

Q. If a star schema has 3 dimensions and one fact:
   – Product with 1000 rows
   – Time with 365 rows
   – Sales Org with 2000 rows
   – Sales Order transactions fact table of 50 rows,
   How many rows are returned in a query of select * with all dimensions joined?

What else can you observe about this star schema?
BI Data Types

NOIR

- **Nominal** – Red, Yellow and Blue
  Just a name with no order or magnitude

- **Ordinal** – 1\textsuperscript{st}, 2\textsuperscript{nd} and 3\textsuperscript{rd}
  An order, but no magnitude

- **Interval** – 1”, 2” and 3”
  An order, a magnitude but absolute zero

- **Ratio** – -2\degree, -1\degree, 0\degree, +1\degree, +2\degree C
  An order, a magnitude and below zero possibilities
## Fact and Dimensions

**What goes where?**

<table>
<thead>
<tr>
<th>Dimensions</th>
<th>Fact</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Nominal</strong></td>
<td><strong>Ordinal</strong></td>
</tr>
<tr>
<td>• Red, Yellow and Blue</td>
<td>• 1st, 2nd and 3rd</td>
</tr>
<tr>
<td>• Just a name with no order or magnitude</td>
<td>• An order, but no magnitude</td>
</tr>
<tr>
<td>• Master Company</td>
<td>• LOB Rank by Margin</td>
</tr>
<tr>
<td>• Business</td>
<td>• Top 10 LOBs by Policy Counts Sold</td>
</tr>
<tr>
<td>• Line of Business</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Slowly Changing Dimensions
What’s the best practice?

- Slowly Changing Dimensions:
  - Type I, doesn’t preserve history
  - Type II, preserve a version of history
  - Type III, Hybrid of Type I and II

- A Type II writes a new row.
- Type II changes let you preserve historical data.
- Implementing Type II changes, after the fact, will require significant analysis and development.
- Type II changes add rows and will significantly increase the database size.
SCD II: An Example

Why have two dates?
Why a null value for Order_Status_End_Date?
Why have an occurrence counter?
A dimension table can translate to the semantic layer in three ways:

- **One to One** - The dimension table column translates to one object in the semantic layer.

- **Traditional Hierarchies** - Multiple table columns like, Sales Organization and Department and Unit, translate into ‘Sales Org Hierarchy’.

- **Temporal Hierarchies** - A hierarchy is generated from the rows within a column, this only happens in a time dimension.
## Time Dimension

### What are the best practices?

- Include yesterday and tomorrow on today’s row
- Store all the date parts
- Store codes and decodes

<table>
<thead>
<tr>
<th>PK_Date</th>
<th>Calendar_Date</th>
<th>Julian_Date</th>
<th>Day_Of_Week_Name_Sunday_Start</th>
<th>Day_Of_Week_Name_Monday_Start</th>
<th>Quarter_Of_Year_Name_Short</th>
<th>Quarter_Of_Year_Name_Long</th>
<th>Month_Of_Year_Name_Long</th>
<th>Month_Of_Year_Name_Short</th>
<th>Week_Of_Year_Number</th>
<th>Month_Of_Year_Number</th>
<th>Start_Of_Week_Date_Sunday_Start</th>
<th>End_Of_Week_Date_Sunday_Start</th>
<th>Next_Calendar_Date</th>
<th>Next_Month_Of_Year_Number</th>
<th>Next_Quarter_Of_Year_Number</th>
<th>Next_Year_Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>01/01/1900</td>
<td>2447893</td>
<td>2</td>
<td>1 Monday</td>
<td>Mon</td>
<td>1</td>
<td>January</td>
<td>1</td>
<td>Q1 1900</td>
<td>N</td>
<td>365</td>
<td>12/31/1900</td>
<td>01/01/1900</td>
<td>01/07/1900</td>
<td>01/01/1900</td>
<td>01/01/1900</td>
<td>03/31/1900</td>
</tr>
<tr>
<td>01/02/1900</td>
<td>2447894</td>
<td>2</td>
<td>2 Tuesday</td>
<td>Tue</td>
<td>2</td>
<td>January</td>
<td>1</td>
<td>Quarter 1</td>
<td>Q1 1900</td>
<td>N</td>
<td>360</td>
<td>12/31/1900</td>
<td>01/01/1900</td>
<td>01/07/1900</td>
<td>01/01/1900</td>
<td>01/01/1900</td>
</tr>
<tr>
<td>01/03/1900</td>
<td>2447895</td>
<td>3</td>
<td>3 Wednesday</td>
<td>Wed</td>
<td>3</td>
<td>January</td>
<td>1</td>
<td>Quarter 1</td>
<td>Q1 1900</td>
<td>N</td>
<td>365</td>
<td>12/31/1900</td>
<td>01/01/1900</td>
<td>01/07/1900</td>
<td>01/01/1900</td>
<td>01/01/1900</td>
</tr>
<tr>
<td>01/04/1900</td>
<td>2447896</td>
<td>4</td>
<td>4 Thursday</td>
<td>Thu</td>
<td>4</td>
<td>January</td>
<td>1</td>
<td>Quarter 1</td>
<td>Q1 1900</td>
<td>N</td>
<td>365</td>
<td>12/31/1900</td>
<td>01/01/1900</td>
<td>01/07/1900</td>
<td>01/01/1900</td>
<td>01/01/1900</td>
</tr>
<tr>
<td>01/05/1900</td>
<td>2447897</td>
<td>5</td>
<td>5 Friday</td>
<td>Fri</td>
<td>5</td>
<td>January</td>
<td>1</td>
<td>Quarter 1</td>
<td>Q1 1900</td>
<td>N</td>
<td>365</td>
<td>12/31/1900</td>
<td>01/01/1900</td>
<td>01/07/1900</td>
<td>01/01/1900</td>
<td>01/01/1900</td>
</tr>
<tr>
<td>01/06/1900</td>
<td>2447898</td>
<td>6</td>
<td>6 Saturday</td>
<td>Sat</td>
<td>6</td>
<td>January</td>
<td>1</td>
<td>Quarter 1</td>
<td>Q1 1900</td>
<td>N</td>
<td>365</td>
<td>12/31/1900</td>
<td>01/01/1900</td>
<td>01/07/1900</td>
<td>01/01/1900</td>
<td>01/01/1900</td>
</tr>
<tr>
<td>01/07/1900</td>
<td>2447899</td>
<td>1</td>
<td>7 Sunday</td>
<td>Sun</td>
<td>7</td>
<td>January</td>
<td>1</td>
<td>Quarter 1</td>
<td>Q1 1900</td>
<td>N</td>
<td>365</td>
<td>01/07/1900</td>
<td>01/13/1900</td>
<td>01/08/1900</td>
<td>01/14/1900</td>
<td>01/01/1900</td>
</tr>
</tbody>
</table>
Time Dimension Usage
How to handle multiple uses of the time dimension

• Use one physical time dimension
• Creates database views to mimic the different uses of the time dimension
• Multiple views allow the dimension to be customized within the cubing tool.

• A little demo…
Reporting Hierarchy Support

What to include to make hierarchies hum?

- Use substring to create a hierarchy when list of values is large
- Make enough levels in hierarchy to limit list of values to 32K or fewer
- Update attribute relationships to avoid full dimension table scans

- A little demo…
Fact Table: Non-Cumulative Measures
What to store?

- Non-cumulative measures cannot be stored in the physical database; they must be calculated at the OLAP presentation point.
- Non-cumulative measures typically have division, % or are ratios like:
  - Gross Margin %
  - GPA
  - Inventory Turn
Fact Table: A Qualifier Matrix

What measures can be aligned to what dimensions?

- Fact qualifier matrix helps plan for implementation and phasing
- Driver of cost and timeline
- A little demo…
Summary

• A data warehouse model is a subset of the BI model
• BI models enable semantic layer tools for reporting and cubing
• Think twice about time dimension
• Embed as many columns in the physical model as possible to allow full downstream usage
Physical Model and Cube Interplay

- Attribute relationships
- Substring 1, 3 to build hierarchy
- Distinct counters in fact table
- Key FK and PK names
- SCD columns
- Analytical groups to address bad data
- Child has only 1 parent
- Low grain and high grain
- Cardinality and aggregate relationship
- Low grain and high matches grain of fact table
- Sort order option on Analysis Services
- Partitions queries and Slice
- MDX
- Distinct counts requires separate measure groups for disk sorting
Call Us!
We’re always good for a free day of consulting.

www.IncisiveAnalytics.com

Christina Rouse
Chief Architect
chris.rouse@incisiveanalytics.com
216.849.8237