Designing a Dimensional Model

Erik Veerman
Atlanta MDF member
SQL Server MVP, Microsoft MCT
Mentor, Solid Quality Learning

Definitions

• Data Warehousing
  A subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of decision-making process.

• Data Marts
  • R. Kimball - “a data mart is a flexible set of data, ideally based on the most atomic (granular) data possible to extract from operational source, and presented in a symmetric (dimensional) model that is resilient when faced with unexpected user queries”
  • “in its most simplistic form a data mart represent data from a single business process”
  Business process = purchase order, store inventory, etc
Definitions

• OLAP = On-line Analytical Processing
  A reporting system designed to allow different flexible analysis in real time, on-line, with data structures designed for fast retrieval, with redundancy included to support performance. **Note:** "On-line" doesn't indicate data from on-line systems, rather on-the-fly

• OLAP Basics
  - **Drill-down** - decreasing the level of aggregation
  - **Drill-up/Roll-up** - increasing the level of aggregation
  - **Drill-across** - move between different star-join schemas using conformed dimensions and joins
  - **Slicing and dicing** - ability to look at the database from different views, e.g. one slice shows all sales of product type within regions, another slice shows all sales by sales channel within each product type
  - **Pivoting** - e.g. change columns to rows, rows to columns
  - **Ranking** - sorting

---

Definitions

**Business Intelligence (BI)**

Forrester definition: A process of transforming data into information and making it available to users in time to make a difference

**Strategic** BI (Examples: Balance scorecard, Strategic Planning)
- Who: strategic leaders
- What: formulate strategy and monitor corporate performance

**Operational** BI (Examples: Budgeting, Sales forecasting)
- Who: operational managers
- What: execution of strategy against objectives

**Analytical** BI (Examples: Financial and Sales Analysis, Customer Segmentation, Clickstream analysis)
- Who: analysts, knowledge worker, controller
- What: ad-hoc analysis
THE Definition

- Dimensional Modeling

The process and outcome of designing logical database schemas created to support OLAP and Data Warehousing solutions

Transactional System Emphasis

- OLTP = On Line Transactional Processing
  - Each transaction must be written to the database in real time, i.e. "on line"

- Data structures must enable consistent and fast writing
  - The best consistency and speed can be achieved if each piece of data is written only once
  - Normalized (netlike) relational schema suits these requests
Reporting Challenges with OLTP

- Schema doesn’t clearly call out subjects, objects, events, states...
- Difficult to prepare reports and analysis views
  - Requires multiple joins
  - Indexes not optimized for reporting
- Models business process, not information
- Levels show only current state, history is not tracked

Data Warehousing, The Solution

- Schema designed with reporting and analysis in mind
- With redundant data, specially prepared for analysis, we can do more:
  - Prepare data over time
  - Prepare aggregates
  - Add data from other sources, not only OLTP
  - Sales value shows much more if we know also market capacity and our market share
Business Intelligence Architecture

- Store Costs & Sales (SQL Server)
- Warehouse Orders & Costs (Oracle)
- Corporate HR & Budget (DB2 & Excel)

Corporate Data Warehouse

- Star-Schema (SQL Server)
- Cubes (Analysis Services)

BI Solutions

Reporting Applications

Dimensional Modeling

- Used by most contemporary BI solutions
  - “Right” mix of normalization and denormalization often called Dimensional Normalization
  - Some use for full data warehouse design
  - Others use for data mart designs
- Consists of two primary types of tables
  - Dimension tables
  - Fact tables
Dimensional Modeling

- **Dimensional normalization**
  - Logical design technique that presents data in an intuitive way allowing high-performance access
  - Targets decision support information
  - Focused on easy user navigation and high performance design

- **(vs.) Transactional normalization**
  - Logical design technique to eliminate data redundancy, to keep data consistency, and storage efficiency
  - Makes transactions simple and deterministic
  - ER models for enterprise are usually complex often containing hundreds, or even thousands, of entities/tables

Dimension Tables

- **Contain attributes related to business entities**
  - Customers, vendors, employees
  - Products, materials, even invoices (attributes!)
  - Dates and sometimes time (hours, mins, etc.)

- **Often employ surrogate keys**
  - Defined within the dimensional model
  - Not the same as source system primary, alternate, or business keys

- **Not uncommon to have many, many columns**

- **Highly de-normalized to reduce joins**
Fact Tables

- Contain numbers and other business metrics
  - Define the basic measures users want to analyze
  - Numbers are then aggregated according to related dimensions
- Fact tables contain dimension keys
  - Defines relationship between measures and dimensions using surrogate keys
- Typically narrow tables, but often very large
- Highly normalized structure
Why Dimensional Modeling

- Logical model is easy to understand
  - Standard framework and business model for end user apps
  - Model can be done (mostly) independent of expected queries
  - Handle changes easy – such as adding new dimensional attributes

- Optimized for performance
  - High performance “browsing” across the attributes
  - Strategy to handling aggregates, leveraging summary tables or OLAP aggregation technologies
  - Logical redundant with base table to enhance query performance
  - OLAP engines can make strong assumptions on how to optimize

- Historical tracking of information
  - Strategies for handling changing dimensions
  - Fact design allows high volume snapshots and transaction tracking

Dimension Tables!
## Dimensions

- Organized hierarchies of categories, levels, and members
- Used to “slice” and query within a cube
- Business perspective from which data is looked upon
- Collection of text attributes that are highly correlated (e.g. Product, Store, Time)
- Shared with multiple fact relationships to provides data correlation

<table>
<thead>
<tr>
<th>State ID</th>
<th>Region</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>East</td>
<td>Maine</td>
</tr>
<tr>
<td>2</td>
<td>East</td>
<td>Ohio</td>
</tr>
<tr>
<td>3</td>
<td>West</td>
<td>Idaho</td>
</tr>
<tr>
<td>4</td>
<td>West</td>
<td>Texas</td>
</tr>
</tbody>
</table>

## Dimension Details

- **Attributes**
  - Descriptive characteristics of an entity
  - Building blocks of dimensions, describe each instance
  - Usually text fields, with discrete values
  - e.g., the flavor of a product, the size of a product
- **Dimension Keys**
  - Surrogate Keys
  - Candidate Business Keys
- **Dimension Granularity**
  - Granularity in general is the level of detail of data contained in an entity
  - A dimensions granularity is the lowest level object which uniquely identifies a member
  - Typically the identifying name of a dimension
Hierarchies

- **Dimension Business Key**
  - Column or columns that identify a unique instance of the business record (not necessarily a unique record in the dimension table)
  - Used in the ETL process to tie fact records with dimension members

- **Dimension Record Surrogate Key**
  - Defines the dimension’s primary key
  - Relates to the fact table foreign key field
  - Numeric data type, typically integer (2, 4, 8 byte)
Dimension Surrogate Keys

- **Surrogate Key Usage**
  - Consolidates multi-value business keys
  - Allows tracking of dimension history
  - Standardizes dimension tables
  - Limits fact table width for optimization

- **Surrogate Key Design Practices**
  - Avoid smart keys
  - Avoid production keys (may change!)
  - The company may acquire a competitor and thereby change the key building rules changed record, but deliberately not changed key
  - Narrow as possible

Dimensions Types

- **Basic Dimension Types**
  - Standard Star dimension
  - Snowflake dimension
  - Parent-Child dimensions

- **Advanced Types**
  - Degenerate
  - Profile or Junk dimensions
  - Role-playing and Outriggers
Role-Playing and Time Dimensions

Changing Dimensions

- Problem known as “Slowly Changing Dimensions” (Ralph Kimball)

- Common changing types:
  0: No change.
  1: Not interested in the previous state. Overwrites value.
  2: Slow changes. Adds new row.
  3: Fast changes. Adds new column
SCD Type-1 UPDATE

- In-Place update
- Restates history, cannot query old value
- Locking and contention possible
- Simple

<table>
<thead>
<tr>
<th>Customer Source</th>
<th>Reseller Dim</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer SK</td>
<td>27</td>
</tr>
<tr>
<td>Customer ID</td>
<td>AW014</td>
</tr>
<tr>
<td>Customer Name</td>
<td>Bike Mart</td>
</tr>
<tr>
<td>Population</td>
<td>Bike Mart</td>
</tr>
<tr>
<td>Customer Type</td>
<td>Business, Warehouse</td>
</tr>
<tr>
<td>Reseller Type</td>
<td>Warehouse</td>
</tr>
<tr>
<td>Product Line</td>
<td>All</td>
</tr>
<tr>
<td>Year Opened</td>
<td>1996</td>
</tr>
<tr>
<td>Year Opened</td>
<td>1996</td>
</tr>
</tbody>
</table>

SCD Type-2 UPDATE

- Track History (versioning)
- Surrogate Keys required!

<table>
<thead>
<tr>
<th>Product Source</th>
<th>Product Dim</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product ID</td>
<td>BK-M82S</td>
</tr>
<tr>
<td>Name</td>
<td>Yukon Special</td>
</tr>
<tr>
<td>Size</td>
<td>44</td>
</tr>
<tr>
<td>Model</td>
<td>All Terrain</td>
</tr>
<tr>
<td>Class</td>
<td>Light Frame</td>
</tr>
<tr>
<td>Product SK</td>
<td>462</td>
</tr>
<tr>
<td>Product ID</td>
<td>BK-M82S</td>
</tr>
<tr>
<td>Name</td>
<td>Yukon Special</td>
</tr>
<tr>
<td>Size</td>
<td>44</td>
</tr>
<tr>
<td>Model</td>
<td>All Terrain</td>
</tr>
<tr>
<td>Class</td>
<td>Light Frame</td>
</tr>
<tr>
<td>Start Time</td>
<td>1/1/2003</td>
</tr>
<tr>
<td>End Time</td>
<td>6/7/2005</td>
</tr>
</tbody>
</table>

//atlanta.mdf
Where to Start for Dimensions..

- Understand dimension hierarchies and drill-paths
- Confirm historically tracked attribute requirements
- Check source data integrity, cleanliness, and completeness
- Review current reports for summarization, roll-up, and grouping

Fact Tables!
### Facts

- **The fact itself**
  - The “measure” that is being tracked
  - Quantity, count, amount, percent
  - Mostly numerical, continuous values
  - e.g., price of a product, quantity sold, number of products in inventory, budget value, count of customers, count of sales, account balance

- **Facts (or measures) can be classified by...**
  - Numerical data type
  - Aggregation type
  - Additive nature
  - Granularity

### Fact tables

- Capture measures/facts
- Association with dimensions
- Some tracking information included

### Fact Table Granularity

- The level of detail of data contained in the fact table
- The description of a single instance (a record) of the fact
  Typically includes a time level and a distinct combinations of other dimensions

- e.g. Daily item totals by product, by store,
  Weekly snapshot of store inventory by product

---

### Different types

- Transactional
- Snapshot or inventory
- Factless
Additive Nature

- **Additive**: Facts that can be summed up/aggregated across all of the dimensions in the fact table (e.g., discrete numerical measures of activity, i.e., quantity sold, dollars sold)
- **Semi-Additive**: Facts that can be summed up for some of the dimensions in the fact table, but not the others (e.g., account balances, inventory level, distinct counts)
- **Non-Additive**: Facts that cannot be summed up for any of the dimensions present in the fact table (e.g., measurement of room temperature)

Advanced: Custom Rollups

<table>
<thead>
<tr>
<th>Financial Statement</th>
<th>Standard</th>
<th>Custom</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profit</td>
<td>13000</td>
<td>1000</td>
</tr>
<tr>
<td>+ Net sales</td>
<td>8300</td>
<td>5700</td>
</tr>
<tr>
<td>+ Gross sales</td>
<td>7000</td>
<td>7000</td>
</tr>
<tr>
<td>- VAT</td>
<td>800</td>
<td>800</td>
</tr>
<tr>
<td>- Discount</td>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td>- Expenses</td>
<td>4700</td>
<td>4700</td>
</tr>
<tr>
<td>+ Infrastructure</td>
<td>1500</td>
<td>1500</td>
</tr>
<tr>
<td>+ Labor</td>
<td>2500</td>
<td>2500</td>
</tr>
<tr>
<td>+ Financing</td>
<td>700</td>
<td>700</td>
</tr>
</tbody>
</table>

- Possible, but requires blending of data with meta data in the data warehouse
Aggregations (Aggs)

- A summarization of base-level fact table records
- Aggregations need to account for the additive nature of the measures, created on-the-fly or by pre-aggregation
- Common aggregation scenarios
  - Category product aggs by store by day
  - District store aggs by product by day
  - Monthly sales aggs by product by store
  - Category product aggs by store district by day
- Common aggregations = Sum, Count, Distinct Count, Max, Min, Average, Semi-additive (Last Child, Last Non-empty Child)

Fact Table Types

- Different types of fact tables
  - **Transactional** – Additive facts tracking events over time
  - **Snapshot or inventory** – Pictures in time of levels or balances
  - **Factless** – Dimensionality relationships
  - **Combinations!**
Transactional Fact Tables

- Most common type of fact table
- Track the occurrence of events, each detailed event is captured into a row in the fact table
- Measures are typically additive across all dimensions
- Common transactional fact table types
  - Sales, Visits, Web-page hits, Account transactions

Snapshot Fact Tables

- Known as inventory level fact tables
- “Snapshot” in time of quantities in stock or balances of accounts
- Time dimension used to identify grain
- Non additive measures across time, but typically additive across all other dimensions
- Common transactional fact table types
  - Inventory levels, Event booking levels, Chart of account balance levels
### Factless Fact Tables

- No measured facts!
  - Are useful to describe events and coverage
  - Information that something has or has not happened
- Often used to represent many-to-many relationships
- Contain only dimension keys
- Common factless fact tables:
  - Class attendance, Event tracking, Coverage tables, Promotion or campaign facts

### Design with Additive in Mind

- Think dimensionally!
- Complex requirements don’t need to be designed with complex queries
- Many times new fact tables can be designed that can answer specific questions, such as date attributes and ranges
Where to start with Fact Tables...

- Identify high-value business process to model (orders, invoices, shipments, inventory)
- Identify reporting grain of the business process
- Identify dimensions that apply to each fact table
- Identify measures that will populate the fact tables

Example business questions to listen for:
- How much total business did my newly remodeled stores do compared with the chain average?
- How did leather goods items costing less than $5 do with my most frequent shoppers?
- What was the revenue comparison of non-holiday weekend days to holiday weekend days?

Resources

- The Data Warehouse Toolkit (Kimball)
- The Data Warehouse Lifecycle Toolkit (Kimball)
- The Microsoft Data Warehouse Toolkit (Kimball)
- OLAP Solutions (Thomsen)
- Microsoft OLAP Solutions (Thomsen, Spofford, Chase)
Simple as that! 😊

What we didn’t have time to cover:

Meta Data in Data Warehouses
Designing tables with SSAS in mind
Indexing strategies
Disk optimization for Data Warehouses
Fact table partitioning strategies
Aggregation tables
ETL Considerations
Data Warehouse Project Planning