Implementing a Taxonomy

A Comparison of Database Approaches

Vignette Content Management Blueprint

White Paper
December 2002. v.1.0
# Table of Contents

- **Introduction** ............................................. 1
- **Taxonomy Approaches** ................................. 1
  - String-Based Approach ................................ 1
  - Dimension-Based Approach ......................... 1
- **Data Approaches** ....................................... 2
  - Editorial .................................................. 2
  - Transactional ............................................ 2
- **Schema Approaches** .................................. 2
  - Single-Table Approach ................................. 2
  - Single-Table Example Schemas ................. 2
  - Multi-Table Approach ............................... 4
  - Multi-Table Example Schemas .................. 4
- **Conclusion** ........................................... 5
- **Appendix A. Detailed Database Schema** .......... 6
Implementing a Taxonomy

A Comparison of Database Approaches

Vignette Content Management Blueprint

Introduction

The concept of a content classification taxonomy is crucial to the proper implementation and maintenance of a content management solution. A content taxonomy provides order to a large volume of content, and allows business users to navigate and manage content more efficiently. This paper addresses two basic approaches for providing taxonomy support in the content management application, and explains how an implementation should be performed. The Appendix provides a sample detailed database schema. This paper assumes that readers have read “Designing an Integrated Content Management Solution: A Taxonomy-Based Approach” and are familiar with various taxonomy descriptions and terminology.

Taxonomy Approaches

There are basically two types of taxonomies. The first is a string-based taxonomy. In a string-based approach, taxonomy nodes are represented by a string that designates the position of a category in the hierarchy. The second type is a dimension-based approach that allows for the categories in the taxonomy to be broken down into discrete areas. A dimension-based approach allows more flexibility in placing content into various nodes represented by the dimension to which they belong. Dimensions are most commonly utilized in a navigation taxonomy to present multiple views of the same content. For example, dimensions in a retail taxonomy may be product type (sweaters, shoes, etc.), gender (male, female), or type (casual, evening, special, etc.). Content items can be classified using nodes in one or all of these dimensions.

String-Based Approach

In a string-based approach, nodes are formed by a separate string that defines the node and its position in the hierarchy. For example, “product > sweaters > women’s > casual”. One advantage of this approach is that it allows for a linear view of the content categories. It also provides flexibility in adding new nodes and ensures that new nodes can be added without affecting other taxonomy constraints. A string-based approach also provides a clearer view of the taxonomy when performing searches. However, there are some drawbacks to the string-based approach. First, this approach forces the taxonomy to be linear by design, and although this helps build in structure, it limits the robustness of the taxonomy. Second, utilizing a string to represent the taxonomy makes it difficult to provide multiple tagging capabilities for specific pieces of content. Lastly, the string-based approach typically does not enforce referential integrity and thus makes it more difficult to maintain the taxonomy.

Dimension-Based Approach

A dimension-based taxonomy approach provides more flexibility in tagging to categories in specific dimensions. It also enforces a more structured approach to the hierarchy of the taxonomy. Dimensions also typically build in referential integrity, which aids in the maintenance of the taxonomy. Some of the disadvantages of this approach involve performance and tagging constraints. Performance
may become an issue if items are tagged across several dimensions and cause SQL joins across multiple tables and complicated queries. Tagging may also become an issue if several dimensions are built into the taxonomy. If a content producer must tag content for multiple dimensions, this approach increases the content tagging work significantly.

**Data Approaches**

Another key area to consider when deciding how to implement the database design to support a taxonomy is the approach used for storing the data that represents the taxonomy. There are two basic types of data approaches: an editorial approach to data in which data will be viewed as mere content that gets published and pushed out to the Web, or a transactional approach in which data is created by transactions.

**Editorial**

Editorial data is the easiest of all to deal with in terms of entering, storing, and displaying because it does not require extensive overhead to ensure transactional compliance. An editorial approach typically aligns well with a generic view of data. Editorial data can use a single-table approach, since a single table can view all the content as basically having the same behavior with the only difference being the taxonomy category to which the content items are tagged.

**Transactional**

Transactional data involves a more complex data model due to the fact that entities will behave differently and have specific relationships to other data entities. The transactional approach favors a multi-table view of the data because it creates a need for transactional overhead on specific tables and a need to create one-to-many views of the data. The data is differentiated by the entities in which it lies and forms a foundation for securing and maintaining transactional compliance.

**Schema Approaches**

Once the taxonomy and data approaches have been selected, it is much easier to decide which schema approach to take. The following lists the various approaches and their affinity to the various schema approaches:

- String-based taxonomy with Editorial Data approach favors MULTI-TABLE
- Dimension-based taxonomy with Editorial Data approach favors SINGLE-TABLE
- String-based taxonomy with Transactional Data approach favors MULTI-TABLE
- Dimension-based taxonomy with Transactional Data approach favors MULTI-TABLE

**Single-Table Approach**

The single-table schema approach utilizes a single table to hold content and typically provides a single table to manage the taxonomy. There are several advantages to utilizing this approach. First, it provides an easy and flexible mechanism for publishing content. Second, it allows greater flexibility for use of taxonomy dimensions. Lastly, it provides faster time to deployment. The disadvantages of this approach include scalability, performance, and documentation issues. Scalability issues may arise if a certain threshold is reached within the content table, depending on the specific DBMS used. Performance issues may also be a concern depending upon how the data is queried and retrieved from the database. Documentation can also be an issue if a developer has a need to understand how to retrieve specific content items.

**Single-Table Example Schemas.** The schemas below show examples of the single-table approach.

**String-Based Editorial**
This is a very simplistic view of how a string-based taxonomy can be implemented in the database schema. The Navigation and Content taxonomy are kept separate. A content item can be assigned to multiple navigation taxonomy strings whereas it can only be associated to one content taxonomy string. This approach ensures that content items are only associated to one node within the content taxonomy, which creates a more structured view of the taxonomy for managing the content.

**Dimension-Based Editorial**

In this example, dimensions located within the NAVIGATION_TAXONOMY and CONTENT_TAXONOMY tables represent the taxonomy. The dimensions are represented by the recursive relationship that allows a node to be a parent of nodes below it and to have parent nodes above it. This approach allows for a more structured view of the taxonomy and the various dimensions that make up the taxonomy.
Multi-Table Approach

Using a multi-table is the classical data model approach and is one that will maintain transactional compliance for data exchange and storage. This approach is generally taken when a client wishes to utilize the database beyond mere content management. Most DBAs will also prefer this approach because it allows them to maintain better control over performance and maintenance of the data model.

Multi-Table Example Schemas. Below are some examples of how the multi-table schema might be approached. Refer to the schema in Appendix A for further details.
In this data model, products, services and companies are all comprised of separate entities. This separation is done for transactional purposes to maintain integrity on the information that is being stored, but there is content in these tables which should also be viewed on the Web Site. In order to display the content on the Web Site, the tables should be associated with the navigation taxonomy nodes, and with the content taxonomy nodes for backend content management tagging. This model uses a string-based approach in order to place new products into distinctive nodes without relying on a structured taxonomy.

The example below shows how a transactional schema might look for a dimension-based approach. Product, Service and Company Info all relate to the Content Taxonomy and provide organization for content around these core components of the schema. The navigation taxonomy is used to specify the contents of specific views of product and services data as it applies to company info.

**Conclusion**

This paper has presented several approaches to implementing a taxonomy through database design, and addressed the pros and cons of the approaches and provided examples for database schema models. The Appendix has a more detailed implementation database schema that provides a greater level of detail and real-world requirements.
Appendix A. Detailed Database Schema

Dimension-Based Editorial

Vignette Corporate Headquarters
1601 South MoPac Expressway
Austin, TX 78746-5776
512.741.4300 Tel
512.741.4500 Fax
888.608.9900 Toll-Free
Email info@vignette.com

Vignette Latin America
305.789.6603 Tel
305.789.6612 Fax
latininfo@vignette.com

Vignette Europe / Middle-East / Africa
44.1628.77.2000 Tel
44.1628.77.2266 Fax
euroinfo@vignette.com

Vignette Asia-Pacific
1.800.800.848 Tel
61.2.9455.5200 Fax
asiapacificinfo@vignette.com

Publication date: June 2002. Vignette does not warrant, guarantee, or make representations concerning the contents of this document. All information is provided “AS-IS,” without express or implied warranties of any kind. Vignette reserves the right to change the contents of this document and the features or functionalities of its products at any time without obligation to notify anyone of such changes.

Copyright 1997-2001 Vignette Corporation. All rights reserved.
Vignette, the V Logo, www.vignette.com, StoryServer, netCustomer and Centerstage are trademarks or registered trademarks of Vignette Corporation in the United States and foreign countries. VGM, VPS, and Vignette Village are servicemarks of Vignette Corporation in the United States and foreign countries. All other brands, products and company names mentioned are the trademarks of their respective owners.