In Kürze

Immer mehr Forschende wenden sich an den SNF


Neues EU-Projekt VIKEF zum „Semantic Web“


Fachgruppen Mitteilungen

DBTA

Datenbanken: Theorie und Anwendung
Chairman: Paul Eisner
Korrespondent: Moira Norrie
http://www.unizh.ch/groups/dbtg/DBTA

Invitation to the SI DBTA General Assembly 2004

The General Assembly of DBTA will take place on 19 November 2004 in Zurich, at 08:30 (directly before the Swiss Informatics Day 2004). The exact location and time will be posted on our website <http://www.dbta.s-i.ch/> as soon as possible.
chairman (Paul Eisner has already served two tenures as chairman, the maximum allowed by the DBTA by-laws).

We are looking forward to meet you at our DBTA General Assembly 2004!

Paul Eisner, Chairman DBTA

Trade-offs in the Design of Code Structures
Robert Marti, Swiss Re

Introduction
The discipline of database design – that is, the development of a database schema for one or more applications – has a prominent role in all textbooks and academic curricula covering database management and information systems. Typically, at least one of the many flavors of Entity-Relationship modeling and the basic ideas of relational normalization are presented. Most recent textbooks also mention multidimensional data models as used in Data Warehousing and On-Line Analytical Processing (OLAP), including the design of so-called star and snowflake schemas. But no standard database textbook known to the author discusses the design and usage of a vocabulary of business terms and codes used to classify the data collected as a result of conducting business transactions (e.g., selling products to clients, buying goods from suppliers, paying salaries to employees etc.). However, the usage of a consistent, well structured and well defined vocabulary of business terms and codes used to classify the data collected as a result of conducting business transactions (e.g., selling products to clients, buying goods from suppliers, paying salaries to employees etc.) is important for analyzing the current state of an enterprise and as an input for the decision making process.

In this short article, we look at the forces influencing the implementation of codes and code structures, namely (1) ensuring maximum stability over time in the face of continuous business change, (2) providing an effective and friendly user-interface, and (3) achieving high query performance. We show some of the alternatives in the design and usage of business terms and codes, and we discuss the pros and cons of the various options.

Definitions
The primary identification key of a business term is called a term identifier, a system-generated number which must not change for the entire life-time of a business term and must not be re-used even after the business term in question has been flagged as outdated.

The business term itself, also called the term name, is a “medium-length” text, which consists of letters, e.g. ‘property’, ‘US Dollar’, etc. (The use of digits or special symbols is also allowed but discouraged.) Such business terms are usually displayed on the GUI of applications, e.g., in drop-down lists.

A code is a shorthand (abbreviation) of a business term. Codes are typically alphabetic, in which case they are often chosen to convey some meaning to the user, e.g. ‘USD’ as a shorthand for ‘US Dollar’, hierarchical, expressing the existence of a taxonomy, that is, hierarchical relationships between the business concepts, where business concepts are represented using a numeric (or sometimes alphabetic) encoding scheme, e.g., ‘i’ for P&C business, ‘11’ for property business, ‘113’ for fire business, and ‘12’ for casualty business (see Figure 1).

In order to deal with codes in a logically distributed environment, their scope sometimes has to be limited, e.g., in order to avoid name clashes. Therefore, codes are assigned to terms within a context, where a context may be established by an application, often a legacy system by a standard body e.g. context ‘ISO’, or a default (standard) enterprise context 1

Codes used in existing applications are mapped to a standard business term 2. At the same time, standard codes, belonging to the default context can be defined for each business term. As a result, the common linkage of (1) an application code and (2) a standard code to a business term identified by a term identifier establishes a mapping from the application code to the standard code.

Figure 1: Taxonomy of Codes

---

1 A context is similar to a namespace in programming languages.
2 This mapping is possible only if certain conditions are satisfied. The precise explanation of the conditions to map codes to standard business terms is beyond the scope of this short paper. The simplest case is a one-to-one correspondence between codes and business terms as is the case in "flat" (non-hierarchical) attribute types such as currency codes and country codes. More complex conditions could e.g. ensure that the hierarchy of codes used in a specific context can be “embedded” in the hierarchy of standard business terms.