

I. Overview

Business users are often interested in reporting on a subset of elements that go beyond what a typical reporting system can handle. A user may have information about certain elements (*e.g.*, products, customers) that are known only to the user — information that is not readily available in a typical data warehouse. For example, a user may have information relating to particular products, such as, *e.g.*, whether certain products are promotional in nature. Such information would need to be maintained in a lookup table defined by the user. This table would comprise a subset of all elements for an attribute within the enterprise data warehouse. This approach provides a highly efficient method for filtering and aggregating data using large sets of user-defined information. This paper provides a technical description as to how this may be performed.

II. User-Defined Sets

A User-Defined Set (UDS) provides an efficient method for analyzing large sets of data. Additionally, a UDS allows a user to incorporate additional information into the data warehouse that may only be relevant (and available) to that user. With the aid of a UDS, a subset of data elements is stored in a separate table, which overcomes the inherent performance limitations of traditional filtering methods when working with large data sets (*i.e.*, using an “in” statement in a database query).

An example of when a UDS would be appropriate is as follows: Category Manager Sherry Smith is interested in tracking the sales of approximately 5,000 stockkeeping units (SKUs), which are associated with various promotions that only she is aware of. She has an Excel spreadsheet that maps each SKU to its own promotion. The promotion information is not available in the data warehouse. However, Sherry would like to use the data warehouse to obtain sales information in order to track the success of the various promotions.

Rather than running a report at the SKU level for all SKUs in the data warehouse, a UDS allows Sherry to define this promotion information within the data warehouse. As a result, this subset of SKUs can be efficiently incorporated into any query without the use of large SQL “in” clauses. It is assumed that an interface has been provided to the users to give them the capability of managing their user-defined data sets (this approach is described in the “Managing User-Defined Sets” white paper).

UDSs will be further exemplified through the following scenario: four users (*e.g.*, Category Managers) belonging to three CPG companies, Suppliers A, B and C, create and manage characteristics within a UDS to access data within a data warehouse maintained by XYZ Retail. XYZ Retail allows the users to access a certain fact table (FACT_TRANSACTION, see below) in its database. It is assumed that the users have the capability of uploading (in a bulk manner) data into one or more tables within XYZ Retail’s data warehouse. The four users (and their respective organizations) are defined in a user lookup table (LU_USER, see Table 1), which can be managed by the Database Administrator (DBA) of XYZ Retail.

Table 1: LU_USER includes users and the organizations they belong to

USER_ID	USER_DESC	ORGANIZATION
1	Bill Coddle	Supplier A
2	Xing Zu	Supplier B
3	Sherry Smith	Supplier C
4	Ray Zafer	Supplier A

With reference to Figure 1, LU_USER has a one-to-many parent-to-child relationship with a user-UDS cross-reference table, USER_UDS_XREF (Table 2), which maps users to one or more UDS IDs. In the present example, users Bill Coddle and Ray Zafer have access to UDS_ID 1, Xing Zu has access to UDS_ID 2...