

Design Tip #67 Maintaining Back Pointers to Operational Sources

By Ralph Kimball

Our data warehouses are increasingly oriented toward tracking detailed customer transactions in near real time. And as Patricia Seybold points out in her wonderful book, *Customers.com* (Times Business, 1998), managing customer relationships means having access to the data from all the “customer facing processes” in an organization.

The combination of keeping the detail behind all the customer facing processes, but at the same time providing an integrated view, presents an interesting challenge for the ETL architect. Suppose we have a typically complex customer oriented business with fifteen or more customer facing systems including store sales, web sales, shipments, payments, credit, support contracts, support calls, and various forms of marketing communications. Many of these systems create their own natural key for each customer, and some of the systems don't do a particularly good job of culling out duplicated entries referring to the same customer. There may be no reliable single customer ID used across all customer facing source systems.

The ETL architect faces the daunting task of de-duplicating customer records from each separate source system, matching the customers across the systems, and surviving the best and most reliable groups of descriptive attributes “cherry picked” from each of the systems. I describe the details of de-duplicating, matching, and surviving in ETL subsystem #8 in my ETL classes and book.

The dilemma for the ETL architect is that even after producing a perfect final single record for the customer, the end user analyst may be unable to trace backward from the data warehouse to a set of interesting transactions in just one of the source systems. The de-duplication and survival steps of preparing the final clean customer master may make subtle changes in names, addresses, and customer attributes that decouple the data warehouse from the original dirty transactions back in the source systems.

The recent demand by the end user community to make all customer transaction detail available in the data warehouse means that we need somehow to carry forward all the original source IDs for the customer into the final customer master dimension. Furthermore, if the source systems have generated duplicate records for the same customer (which we find and fix in the ETL pipeline), we need to store all of the original duplicate source system IDs in the customer master dimension. Only by maintaining a complete set of back pointers to the original customer IDs can we provide the level of trace-back service that the end user analysts are demanding.

I recommend creating a single cross reference table to hold all the original customer IDs. This table has the fields

Data Warehouse Natural Customer Key
Source System Name
Source System Customer ID

The Data Warehouse Natural Customer Key is a special natural key created by the data warehouse! We need such a permanent, unchanging key in the data warehouse master customer dimension to unambiguously identify Slowly Changing Type 2 versions of a given customer.

This table can be queried directly, constraining the Data Warehouse Natural Customer Key, or by joining this field to the same field in the master customer dimension. In both cases, the Source System fields will give the complete list of back pointers.

This design has the advantage that simply by adding data rows to our little cross reference table, it flexes gracefully to handle messy duplicated versions of customer IDs in the source systems, as well as the incorporation of new source systems at various points in time.