Star Schema, Dimensions, Facts, Surrogate Keys

There are abundant resources available covering data warehouse concepts. This is a general overview of some of the basics in an attempt to relate these to fundraising data.

Data warehouses use a combination of dimension and fact tables as part of the structure (schema) of the database.

Facts normally translate as numbers, ex. Gift Amount, Pledge Amount, Fund Balance. Tables containing facts are often “thin” – containing only the database keys and numbers.

Dimensions are descriptors used to provide points for amalgamating the numbers (or facts). ex. Constituent Type, Donor Interest, Gift Type. Dimension tables are often “wide” and contain a key and number of words that describe data.

Two other techniques used in a data warehouse are star-schemas (the way the tables are related to each other) and surrogate keys. We’ll look at surrogate keys first and show how these would work in a star schema.

Consider the idea of a surrogate key used to relate solicitors to prospects.

- The solicitor has an id number in the database
- The prospect has an id number in the database

In the transactional database, this information is routinely stored in 2 tables. When the need arises to assign a prospect to a different solicitor, most systems only record and keep track of the current solicitor of the prospect. Systems do not typically record the history of different solicitors assigned to the prospect over time which is where a data warehouse and surrogate key can help.

In this case, the surrogate key for the table we create in the data warehouse to record the history of the relationship is based on columns that uniquely describe this relationship.

Thus, every time a prospect is assigned a solicitor or a solicitor is unassigned, a row is created in the data warehouse. The row has the following information:

- Solicitor ID
- Prospect ID
- Date Assigned
- Date Unassigned

The table below reveals how as we add rows to the table the details begin to “explain” the manner in which the relationship has changed over time; i.e. Solicitor 1 was assigned to Prospect ID 1 within a date range then Solicitor ID 2 was assigned to the same prospect and is currently assigned as there is no end date reflected for the assignment.
The IDs and dates are the dimensions that describe the relationship. When we relate this table to a “fact” table with gift information we can now answer questions such as which Solicitor raises more money with which donors?

As the data model becomes more complex, we can use these concepts to add additional analytics including, for example, in what department or program area was a solicitor consistently more successful and with which group of prospects?

This clearly is significantly over-simplified. With some creative thinking, however, we can quickly move beyond the bounds of “conventional” reporting to add additional insight and value to our data – and move that data towards information and knowledge – so action can be taken to change the direction of the organization if necessary.

Example of a simple star-schema illustrating how dimensions and facts tables correlate in a data warehouse:

With this type of table structure one can quickly look at a donor’s giving history, when they were assigned to a solicitor, and what where their interests on those dates.

Note that the interest description (not a code) and the solicitor name and type (again, not a code) are included in the tables. Such structure eliminates the need to decode these details in a report by having complex joins to additional tables - all of which makes report development much simpler.

Some of this structure does violate concepts of database normalization, (such as the duplication of the gift type description in multiple tables), yet this is a decided benefit of using a data warehouse.

Disk space is very affordable with the offset being that a properly designed data warehouse can significantly reduce the amount of work.

In the above example it is readily seen how easy it would be to add additional fields and descriptions to the tables. This needs only be done once and the “new” information is exposed and consumable in any of our reports and structures that reference the data warehouse.
In a similar fashion, if rules for solicitor type change, the only necessary change is to the dimension table for solicitor, (since this is a table derived from our base system); business rule changes are then reflected in every piece of information that references this table. This reduces work time from having to modify business rules and change query logic in all of our reports; another example of how data warehouse use can significantly decrease maintenance.

Note that organizations typically end up with hybrid models in a data warehouse. In the purest sense, one does not mix dimension and fact tables. But in a practical sense, most of us are not data architects and would not be able to design and work with these concepts as efficiently as the theorists.

About the Author

Brian Dowling has worked at a number of non-profit and higher education organizations in both the United States and Canada where he gained understanding, knowledge and perspectives of managing in small, medium and large shops.

This experience included multiple system conversions, web site development, budgetary and financial responsibilities, operations management and more.

He has consulted in the United States, Canada and Australia, written numerous articles and presents regularly at conferences.

He invests his spare time in SupportingAdvancement.Com and is an amateur musician.

About SupportingAdvancement.Com

SupportingAdvancement.Com is a high traffic web property with the mission of providing free resources of interest to those working in the fundraising profession. Efficiency is crossing the bridge quickly. Effectiveness is deciding if you need to cross at all. Hopefully, you'll find some resources and ideas here to help you be more efficient and effective.