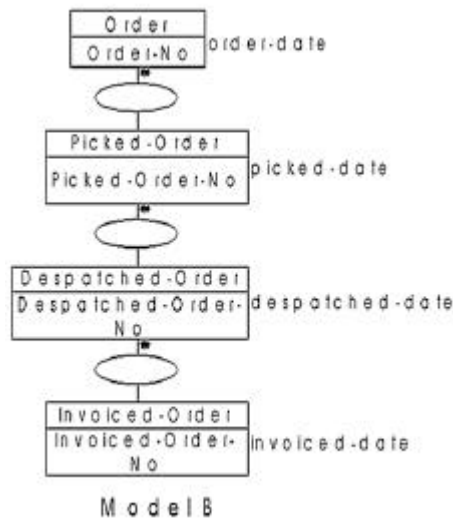
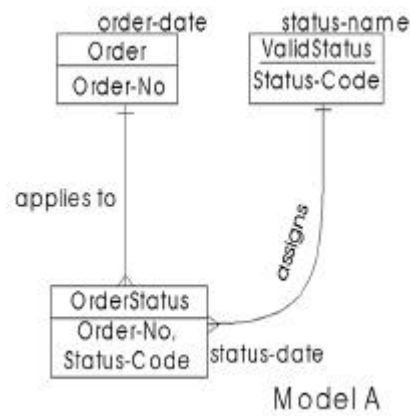


Thoughts about Data (5): Using Entity Models to Guide the Loading of the Data Warehouse

by Mike King

(Read the left column completely before the right column - it seems easier to read in two columns.)



In an article in Datamation (1995), Dr Richard Hackathorn recommends concentrating on how information flows if you want to be successful with a data warehouse.

He classifies the information flows into Inflow, Upflow, Downflow, Outflow and Metaflow.

Inflow is defined as the process of consolidating operational data from diverse transaction processing systems, into the data warehouse. This is done regularly (perhaps daily).

During this process, the data from each source has to be cleansed and rationalized, denormalized and summarized, time stamped, and so on.

The following discussion gives an example of one of these processes, the rationalization of the data, and shows how the transformations may be defined in terms of the entity models.

Consider the business requirement that a record must be kept of the different statuses that a Customer Order may cycle through, including the date on which an order acquired each status.

The valid statuses for an order, and the valid transitions between statuses, are shown in the entity state diagram. According to this, each order must progress through a fixed sequence of states.

It is recorded first. As soon as staff are available, the order items are picked of the shelf. The order is then dispatched. After dispatch an invoice is sent to the customer. Eventually, perhaps 10 years later, the order is deleted from the database.

Two different business departments which handle customer orders may come up with radically different entity models (Models A and B) both of which do a correct and complete job.

Model A would be the result of the perception that each order progresses through many valid statuses, and each valid status applies to many orders. It enables every status of every order to be recorded together with the date of acquiring the status.

Model B would result from the perception that each picked order is an order, each dispatched order is a picked order, and each invoiced order is a dispatched order. It is as correct and complete as Model A (in terms of the limited stated requirements), although it lacks the stability.

Because the two models that correctly represent exactly the same business situation are radically different, and yet the business data from all departments needs to be collected into the same data warehouse, some rationalization will have to be done.

The software that migrates the data from the operational systems into the warehouse will have to transform one of the structures to the other, or both to a third form.

Alternatively, the data could be kept separate in the warehouse, but the software that extracts and presents reports must then have the intelligence to combine and analyse data into the same report from 'competing' structures.

The first alternative, leading to a rationalized warehouse, is the best.

In the case of transforming Model B data to Model A structures, the transformation would be relatively simple.

Each Model B Order would become a Model A Order.

Each Picked Order would become an Order Status, with the picked-date being stored as its status-date, and the Status-Code being set to 'Picked'.

Dispatched Orders and Invoiced Orders would be transformed in the same way.

It is often stressed that the metadata (which is kept up to date by the Metaflows) is an essential part of data warehousing.. In this simple example, the metadata would consist of entity models A & B, the model of the warehouse, and the definition of the transformations to be applied when loading the operational data into the warehouse.