The Data Organization

1251 Yosemite Way
Hayward, CA 94545
(510) 303-8868
info@thedataorg.com

Generalization - A Data Modeling Problem

By Rainer Schoenrank

Data Warehouse Consultant

June 2018

Biography

Rainer Schoenrank is the senior data warehouse consultant for The Data Organization. He has degrees in physics from the University of Victoria and computer science from the University of Victoria and California State University Hayward. He has built data warehouses for clients such as Pacific Bell, Genentech, GE Leasing, SGI, PPFA, Brobeck, BofA, Clorox, Leapfrog and Intuitive Surgical. He can be reached at rschoenrank@computer.org.

Table of Contents

1. INTRODUCTION

The business semantics are very complex—far too complex to explain clearly and reliably in a natural language like English. The functional silos of the business units and the levels in the organization chart use different terms to identify the same business data object (BDO) at various states in its processing lifecycle.

Given the name of a BDO, there are synonyms of the object that are also used by the business, for example, the BDO of employee has alternate names such as associate, sales agent, service provider, customer account rep, sales person, etc. All of these terms are interchangeable depending on the whims of the business.

Also, there are words that represent subsets of an entity, for example, job and position. Position is a structure that contains all of the attributes required by Human Resources for creating, budgeting, hiring, filling, etc., of a place in the business organization. Job contains an outline of the skills and responsibilities for a position.

To create a robust logical data model, the meaning of each table should be generalized to cover all the synonyms and subtypes of that concept and include all of the subset meanings of that concept.

The issue of language in database specification is analyzed by Kent in “Data and Reality” and by Cory Doctorow in MetaCrap.

2. UNDER GENERALIZATION

Given the problems of describing a data model in a natural language, the first issue is the problem of under generalization. That is, a single concept is stored more than one table, for example, the data model contains a table for employee, a table for service provider, etc.

An example of under generalization is the DISPOSITION Data Model #4 by Data Blueprint, Inc. shown in the diagram below.

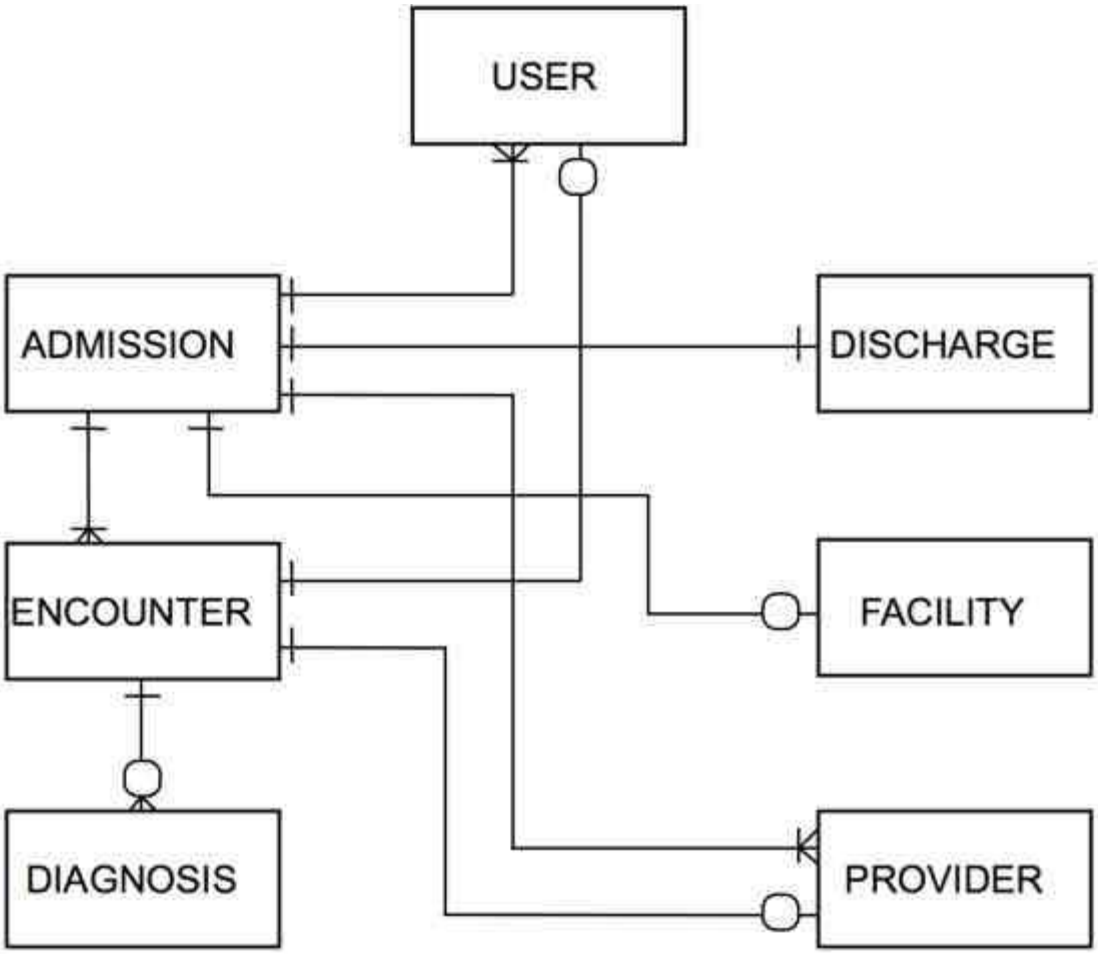

Figure 1. Data Blueprint’s Disposition Data Model

From the Data Blueprint documentation:

Model Purpose Statement: This model codifies the official vocabulary to be used when describing disposition related organizational concepts:

- ADMISSION Contains information about patient admission history related to one or more inpatient episodes
- DIAGNOSIS Contains the International Disease Classification (IDC) of code representation and/or description of a patient's health related to an inpatient code
- DISCHARGE A table of codes describing disposition types available for an inpatient at a FACILITY
- ENCOUNTER Tracking information related to inpatient episodes
- FACILITY File containing a list of all facilities in regional health care system
- PROVIDER Full name of a member of the FACILITY team providing services to the patient
- USER Any user with access to create, read, update, and delete DISPOSITION data

Observations:

The process that stores its data into the data model is when the patient visits a health care facility in order to undergo one or more procedures (a series of encounters). For this process, the descriptions of the tables are:

- user – nurse or clerk – employee
- admission – the beginning of a series of encounters – the beginning event of the patient process being captured
- encounter – patient meets with provider – an event of the patient process being captured
- discharge – end of a series of encounters – ending event of the patient process being captured
- facility – location where the encounter takes place
- provider – doctor or nurse practitioner – employee
- diagnosis – attribute of the encounter

There are some tables missing from the diagram:

- Patient – the person (who) that initiated the encounters
- Calendar – when did the event (admission, encounter, or discharge) occur

Admission, encounter and discharge should be generalized to a single table containing all the patient process events.

User and provider should be generalized into a single table containing the employees of the healthcare system.

When the generalizations are made, the diagram looks like:

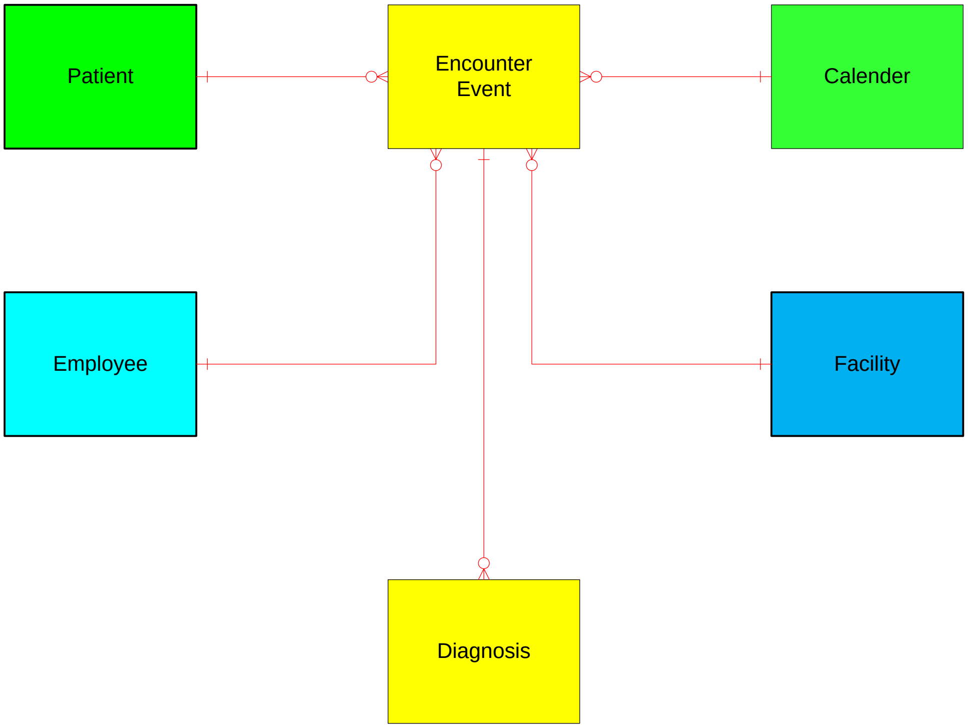

Figure 2. Patient Visit Data Model

Where an encounter event is one of admission, encounter or discharge and employee is one of doctor, nurse practitioner, nurse or clerk.

3. OVER GENERALIZATION

The second issue is the problem of over generalization. To understand over generalization, we need to look at a Kimball star schema data model first. The diagram below shows the star schema for making sales event measurements on a generic sales process.

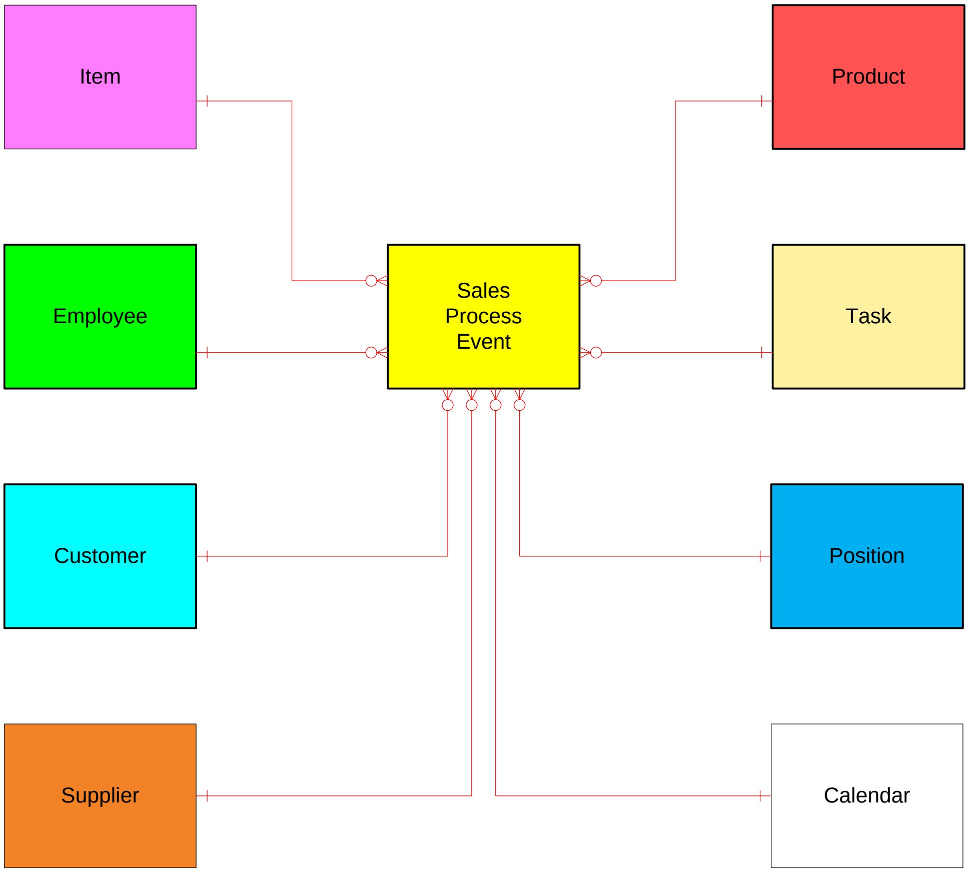

Figure 3. Generic Sales Process Conceptual Data Model

The table at the center of the diagram is the fact table (Kimball) or the measurement table (Sales Process). The arms of the star schema are the master data dimensions shown in the table below:

Master Data Dimension	Description	Zachman Question
Employee	provides labor to the business	who
Customer	sends money to the business for items	who
Supplier	receives money from the business for goods or services	who
Item	inventory item delivered by the business to the customer	what
Product	product ordered by the customer	what
Task	the step in the sales process that the event measures	how
Position	location of business unit within the business	where
Calendar	the date of the sales process event	when

Table 1. The Description of The Star Schema Dimensions

With over generalization, the master dimension tables follow the Zachman categories and several different concepts are stored in a single table (supertype entity). An example of over generalization is found in the ORACLE ERP application that uses the table PARTY (i.e., people we do business with) that will hold customers, employees, and suppliers. The diagram of this type of data model is shown below.

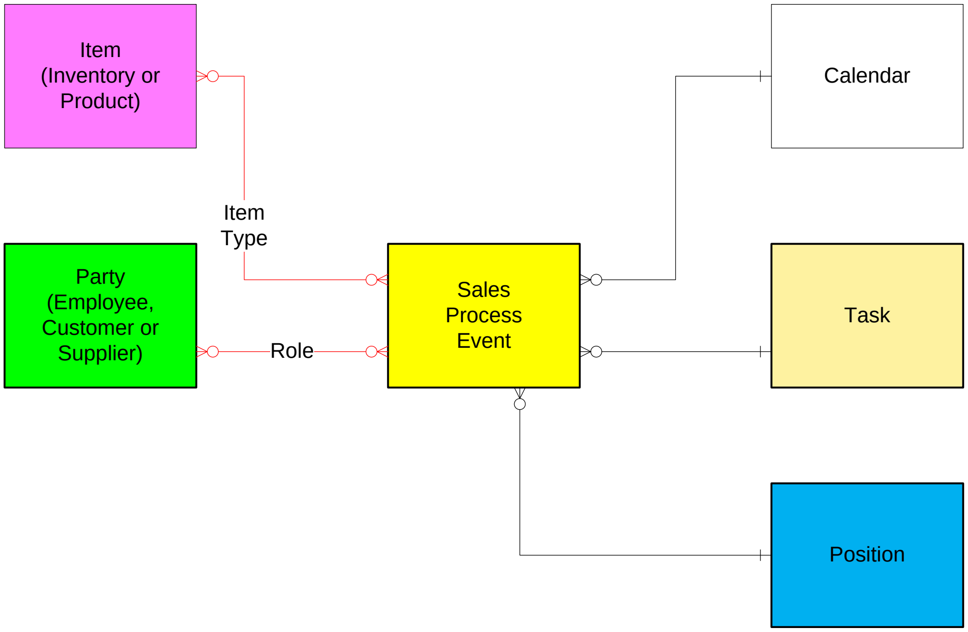
Item (Inventory or Product)
Calendar
Item Type
Party (Employee, Customer or Supplier)
Sales Process Event
Task
Role
Position

Figure 4. Over Generalized Sales Process Conceptual Data Model

This type of over generalization compares badly with the star schema as shown in the table below.

Design Criteria	Supertype Entity	Master Data Entities
Entity structure	very complex	simple
Relationship Type	many to many	one to many
Design Documentation	complex	simple

Table 2. Difference in Design Criteria

Also, this type of over generalization violates the Principle of Orthogonal Design.

The sales process event is an attribute of each of the master data tables and the master data key should appear in the key of the event table. This is not possible in the over generalization since the identifiers for employee, supplier, customer, item and product all appear in the tables that are required to resolve the many to many relationships. The removal of the identifiers for customer, employee and supplier from the sales process event table would cause problems with uniquely identifying the sales events. Because the master data identifiers are not in the sales process event table, this table may suffer from duplicate data rows.

The processing to retrieve data from the database is much more complex than from the master data entities. This added complexity will show down the processing and call for changes to the DBMS hardware and software to get rid of the delays. Examples of the processing are shown in the table below:

Use Criteria	Supertype Entity	Master Data Entities
Entity administration	difficult to organize table partitioning to group customers	straight forward
Read the entity	select * from Party where relationship type = 'Customer'	Select * from Customer
Entity Subtyping	Subtype is in relationship to business measurement not in supertype entity Requires two fields with complex business rule	requires one field with no business rule
Read the entity by subtype	Select * from Party inner join Sales Party Relationship on Party Id where type = 'Employee' and subtype = 'Full Time' may result in duplicate party count	Select * from Employee where subtype = 'Full Time'

Table 3. Differences in Data Usage

4. SUMMARY

With generalization in modeling data, Albert Einstein's maxim needs to be kept in mind at all times:

> "Everything should be made as simple as possible, but no simpler."

5. REFERENCES

Aiken, Peter, Data Modeling Fundamentals, Data Blueprint, Glen Allen, VA, 2018

Codd, Edgar, A relational model for large shared databanks, Communications of the ACM, vol 13, number 6, June 1970

Date, C.J., The Relational Data Dictionary, O'Reilly Media, Inc., Sebastapol, CA, 2006

Kent, William, Data and Reality, North Holland Publishing, New York, 1978