

4/27/2012

Code Data Management

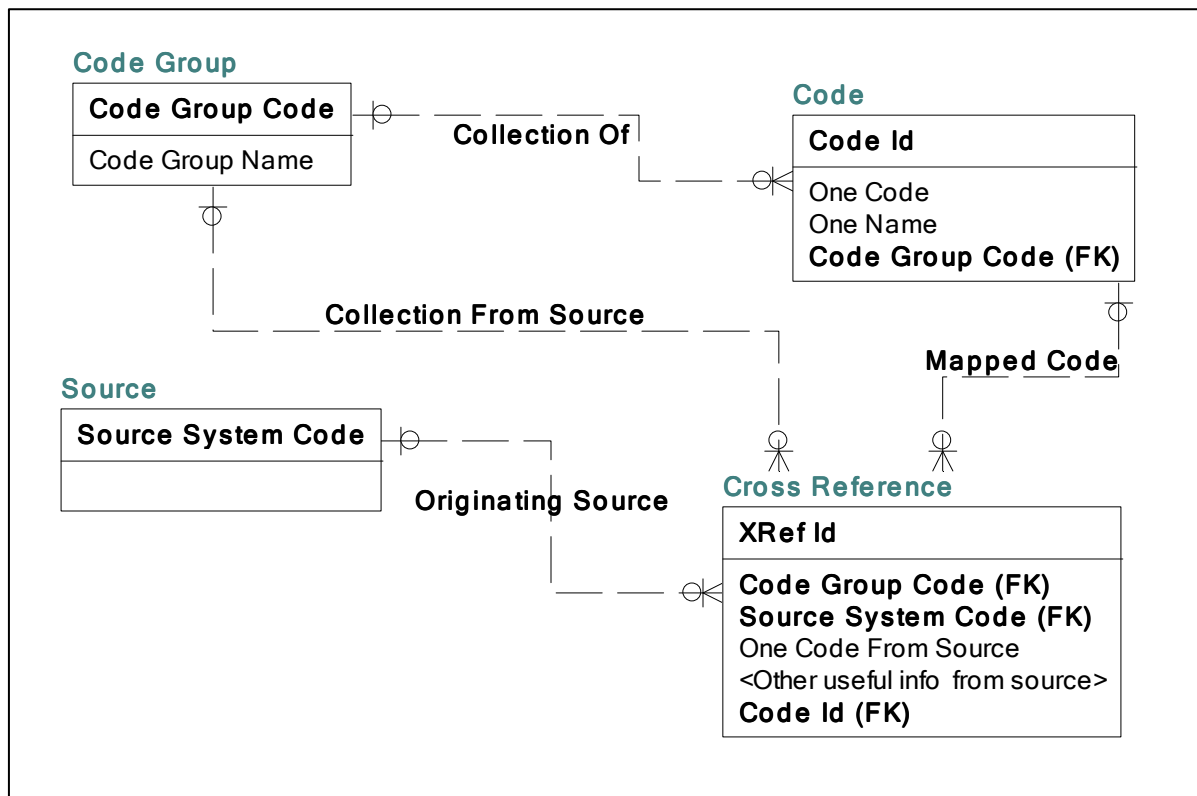
Every enterprise deals with a lot of codes. Some industries a little more than some others. Regardless, we all encounter different ways of handling so many code sets and the many flavors, not minimizing the various sources of this information. Given that the value of codes is well understood, the question is what the right way to manage it is what? Build or buy, which is always a poser. Considerations of cost, time to implement, maintainability, adaptability etc are all good questions that everyone faces.

For the purposes of this article, let us say the decision is pending given the code management comes usually as part of a larger deal (like going to buy a saddle and being forced to buy a horse) like MDM.

3NF is a challenge by itself to an extent to get it right – what with the large number of look up and reference data that potentially play a role. I am sure everyone has a favorite way of handling this beast – below is what I do normally.

Disclaimer: I am not divulging any client secrets as this is a design I adapt everywhere to the extent possible

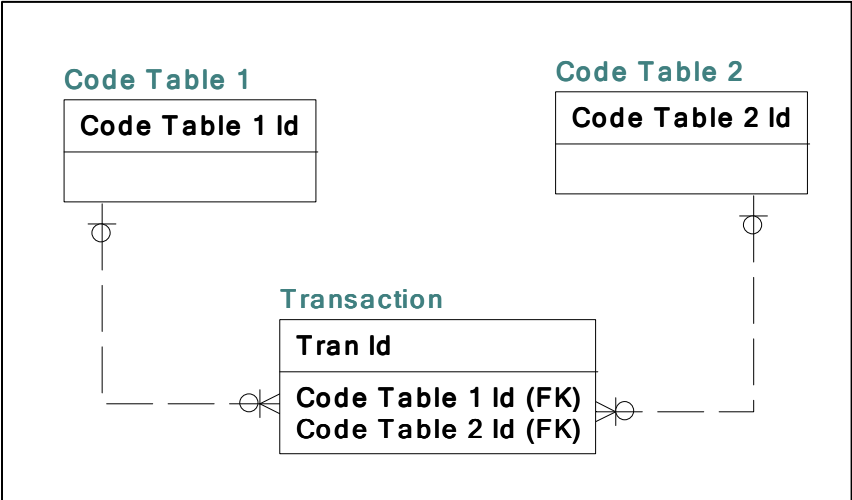
Simple Code Management



What do you do differently that will benefit this design? The other question would be – to model this in ERWin was fine – I would enable all the lookup tables individually as logical only and drive the physical

relationships via these code tables (see below) – how to something like this in TOAD? TOAD seems a lot more restrictive in sense of handling LDM, PDM.

Logical Code Relationships



In the PDM, these two tables don't appear – only the code management tables (I hear some of you say duh).