

Grounded Theory - Open Coding Part 3

It is a categorizing of the data, in fact, I think I said in an earlier session, that I was quite kind of easygoing about the use of the terms, like code and index and category and so on, I use them interchangeably and I do use them interchangeably. In Grounded Theory, the term category is used a lot, so that what you're doing to try to categorize the data, the codes are categorizing and that's the language that the grounded theorists are using. And hence this categorizing approach is a way to try to bring together everything that's about the same topic, that's doing the same thing, representing -- it's coding. Where does it come from? Well, you've got to name the category you identify and decide what it's about. There is a big debate in the literature about this as well. Some people argue that grounded theorists argue start with entirely open mind, that you don't come with any kind of pre-existing theory at all, unlike last week [analytical] induction where I said you do. In Grounded Theory, you start with nothing, a blank slate if you like or a tabula rasa. Others would say, you can't do that, and there are others in between who say, you do it at a certain stage. And I think a lot of debate between, for example, Glaser on one hand and Charmaz on the other and Strauss and Corbin in the middle of somewhere, is about just where you draw that line.

So, I think at some stage, there's no reason why you can't bring in theoretical ideas, it may not be at the very beginning, you got to leave a bit of an open mind to begin with, but certainly Glaser says this and Strauss and Corbin certainly do as well, that at a certain stage of the analysis, maybe not early on in the interviewing, but at a certain stage of the analysis you can begin to bring in theoretical ideas and that's perfectly legitimate for a grounded theorist to do that. You don't have to keep the theory completely aside, and as I say later on it's actually impossible to do that anyway. So bring in ideas from the literature, but try not to do it too quickly, try not to prejudice yourself into certain directions by that, although that is hard not to do.

Another source which Strauss and Corbin mentioned is this in vivo idea. I've got an example of that coming up later on. In vivo means, these are kind of ideas that come from the participants themselves from the settings you're investigating. In vivo, you know, the Latin for in life. So it's their concept, it's their conceptualizations, maybe even their term, their word you're using to represent what's going on here. So, it's very directly a form of understanding, a form of interpretation used by your participants and you're simply using that -- not the words they use perhaps, but nevertheless the concepts they're using. Sorry, here's the example, didn't realize it was -- I didn't read lower down. This comes from Strauss and Corbin, this particular example. I should say Glaser and Strauss's early work, the work they did that started Grounded theory, was in hospital settings, they were looking at nursing in hospital settings. And here's an example from that setting, the so-called tradition bearer. The nurse on the ward who inculcates new nurses into the rules of how you behave and how you act and so on, on the ward.

In fact, it turns out this is pretty true of a lot of organizations, there's always somebody whose job it is to inculcate the new people with how we do things in this department or on this ward or whatever. So, tradition bearer is the name for a code and things that that person is doing are therefore coded as tradition bearer or tradition bearer's activities or whatever, or reference to tradition bearer. So this is the in vivo code. Now I said earlier on, I talked about dimensions, here I want to talk a bit more about what that is and this again, it comes particularly from Strauss and Corbin. They talk about the way that codes

categories can be dimensionalized. As you collect what people are saying about some particular issue or some particular phenomenon that you've coded the same way, you begin to realize that those different examples have slight differences, there are different properties about them. In their book, in Strauss and Corbin's book, they talk about the way that color has properties, it can have different hue, tone, shade, intensity and so on. Different dimensions of the thing that is colored, can be dark and light, and it can be large and small and so on and so forth.

In the same way they say, your codes can have different dimensions, they can be, you know, about the same thing, but slightly different in certain ways, in certain respects. And the example they give is that watching, so watching somebody, and you could imagine, I don't know, this maybe in a restaurant, the chief, you know, cook might be watching over, the trainee cook, the trainee chef and watching what he or she is doing. And that watching can have things like frequency, duration, extent, intensity, and so on, those kinds of things. So it could be, you're watching for a long time, watching for a short time, watching casually, watching intently, etc. Information passing, another idea has amount of time, manner of passing and so on. Once you start thinking like this, you realize that actually we're talking here about types of, kinds of settings for, reasons for, you know, precursors of, all kinds of things like this, you can start to distinguish the codes from each other. So they may be about the same thing, may all be about watching, but it can be different kinds of watching, for example, or watching in different circumstances for example.

So this is an important aspect of this kind of process of open coding, is to begin to dimensionalize the codes and realize they can be about these different aspects of the same thing. I should say that if you're using software or if you're doing something like template analysis, that idea is very well captured by having subcodes. So, in the software, things like [in vivo], you have subcodes or children codes, children nodes actually of course in in vivo, we saw that when we practiced with the software. In template analysis, you have a hierarchical coding system and the subcodes can be the dimensions of the main code. So, you know, having that kind of hierarchical approach is a very good way of capturing dimensionalization.