

I494 Lecture Notes

Class 9 – October 25, 2010

In this class we discussed data models, which I hope is mostly just refresher. There are some key questions that need to be addressed to do an excellent job in documenting data requirements and design. These questions include:

- What data is needed?
- Where does it come from?
- Who creates the data? Where does it come from?
- How are data related?
- For screens:
 - Who accesses?
 - How often is it accessed?
 - Are there different modes for engaging with the data?
 - Read only
 - Update
 - What about audit trails?
- For reports:
 - How often are they generated?
 - Who can see the reports?
 - Is there a history requirement?
 - Paper, or electronic?
 - Raw data, or formatted?

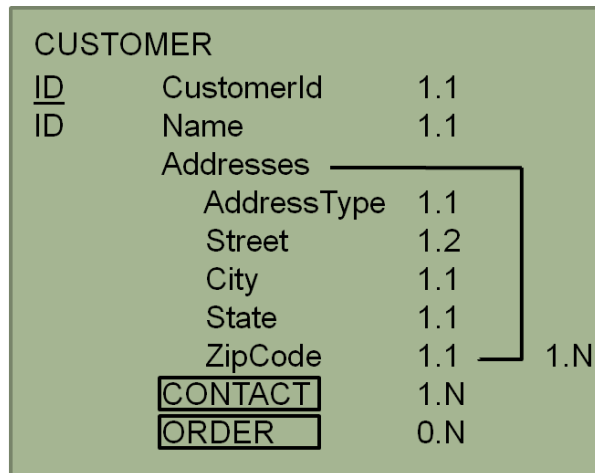
In order to capture the requirements for data management, and to develop good designs I recommend two different conceptual data models. The first is a semantic data model, which is useful for object-oriented systems, relational systems, or even simple file systems. The second is the entity-relationship model, which can be used to support the same as the semantic model. Once you use each for a while, you can even combine the ideas.

I want to stress that data modeling is a process. For me, the process is what matters, because a good process will almost always yield a good model to implement. You need to practice developing models to develop a bag of patterns that you can re-use over and over. There are not that many patterns, and often you are simply changing attribute and table names.

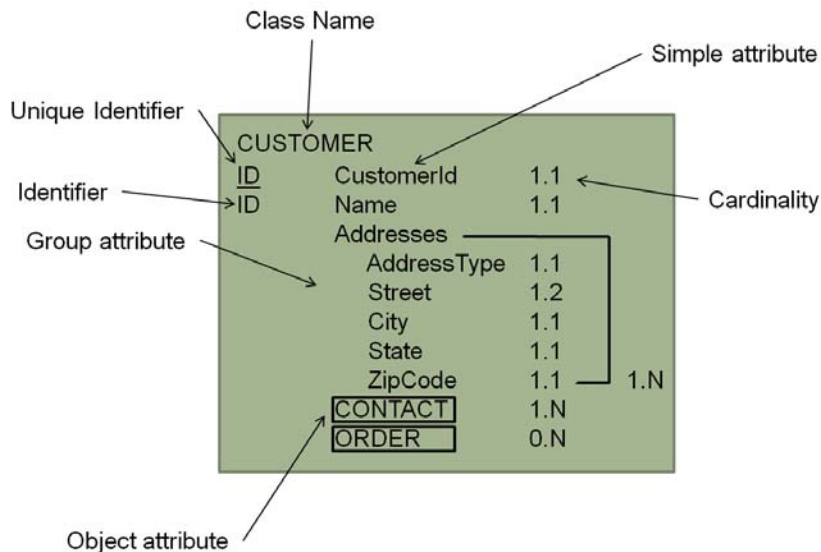
Semantic models use the following key elements:

- Semantic Classes – types of things
 - A named collection of attributes
- Semantic Objects – an instance of a class
- Attributes
 - Simple
 - Group
 - Object
- Cardinality
- Identifiers

Here is the example we used in class:



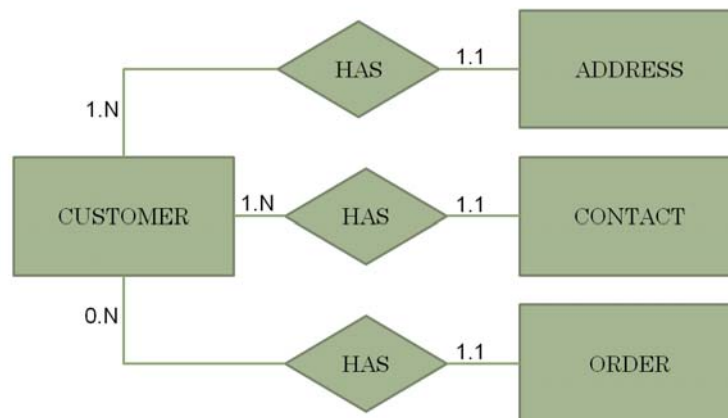
Here is a diagram that shows where everything is:



Entity-relationship models use the following key elements:

- Entities – types of things
- Relationships – how things are related
- Attributes
- Cardinality
- Identifiers

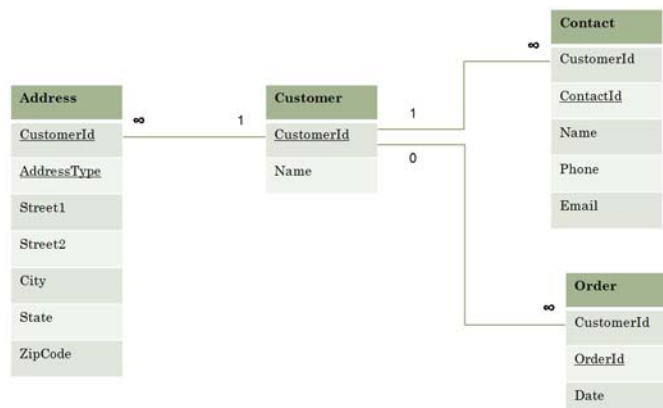
Here is the example as an entity-relationship model:



Relational models use the following key elements:

- Table
- Attributes
- Keys
 - Primary
 - Foreign
 - Calculated
 - Composite

Again, here is the example as a relational model:



An interesting question to consider is how relational databases actually handle cardinality constraints. We talked about the following examples:

- 1.1
- 0.1
- 0.N
- 1.N
- N.N
- 3.5

We went through some in-class exercises to dust off the data modeling skills. Finally, I described what I thought were good choices for a design to include, in terms of data documents:

- Examples of reports and screens
- Conceptual data model
 - Semantic
 - Entity-Relationship
- Implementation data model
 - Relational
 - Structured XML
 - File
- Data dictionary
- CRUD Matrix