

The ER model and the semantic model have both similarities and differences. They are similar in that they both are tools for understanding and documenting the structure of the users data. They will strive to model the structure of the things in the users world and the relationships along them.

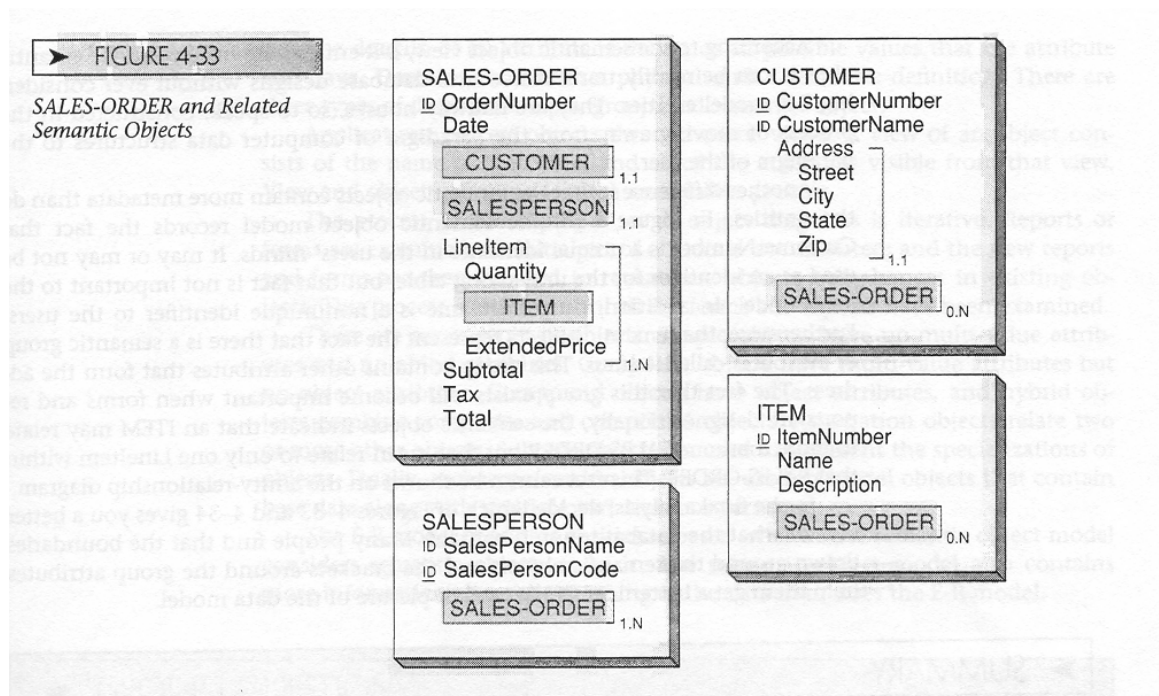
The principle difference between the two models is one of the orientation. The ER model sees the concept of *entity* as basic. Entities and their relationships are considered the atoms, if you will, of a data mode. These atoms can be combined to form what the ER model calls *user views*, which are combinations of entities whose structure is similar to that of semantic objects.

The semantic object model takes the concept of *semantic object* as a basic. The set of semantic objects in a data model is a map of the essential structure of the things that the user considers important. These objects are the atoms of the users world and are the smallest distinguishable units that the users want to process. They may be decomposed into smaller parts inside the DBMS, but those smaller parts are of no interest or utility to the users.

According to the semantic object perspective, entities, as defined in the ER model, do not exist. They are only pieces or chunks of the real entities. The only entities that have meaning to users are, in fact, semantic objects. Another way to state this is to say that semantic objects are *semantically self contained or semantically complete*.

According to the semantic object view, ER entities are unnecessary. Semantic objects can be readily transformed into database designs without ever considering ER model entities.

An other difference is that the semantic objects contain more meta data than do the entities.



In **figure 4-33**, the semantic object model records the fact that CustomerNumber is a unique identifier in the users minds. It may or may not be used as an identifier for the underlined table, but that fact is not important to the data model. In addition, CustomerName is a non unique identifier to the users. Further more, the semantic objects represent the fact that there is a semantic group of attributes called *Attributes*. This group contains other attributes that form the address. The fact that the group exists will become important when forms and reports are designed. Finally, this semantic objects indicate that and ITEM may relate to more than SALES-RDER but that it can relate to only one LineItem within that SALES-ORDER. This fact cannot be shown on the entity relationship diagram.

IN the final analysis, the boundaries drawn around the semantic objects are the brackets around the group attributes help them get a better idea of the overall picture of the data model.