Models attempt to represent the real world in a formal system which can be manipulated to control or predict the behavior of the real world.

In a computer system, we have some limitations in the tools, so we have to fit the problem to the tools.
Entity:

- Entity: A thing that acts or is acted upon -- look for a simple noun.
- The entity in the model is not always one thing in the reality - same person can play the role of "stockholder", "employee", "customer", etc.
- Multiple things in reality can be one entity - a sports team buys a product as a customer in the model, even tho the team is made of many people
  - The concept of a "lawful person" - church, company, team, organization, etc.
Attributes

- Attributes are the properties we wish to model about an entity.
- “An entity is the sum of all its attributes” -- Leibnitz
- .. but we cannot record them all, so we have to pick the important ones
- Here is where we worry about scales, encodings, measurements, constraints, etc.
- Domains: Where the values for the attributes come from.
- We usually encode them -- i.e. we use a color code and not an actual color in the database.
  - One attribute can have many scales
  - Color = Land Color number, Pantone number, etc
  - Temperature = Celsius, Fahrenheit, Kelvin, etc.
- This is a set of values and can be defined by intention (list) or by extension (rules).
It is sometimes hard to tell an attribute from a value.

If I have a table of Sales, is the month of the sale a value or an attribute?

It often depends on your model ...

Quick heuristic: does this thing have to belong to something or is it physically separate?
• Relations put the entities into a system
• Some relationships are important because we use them in the model to create actions in the real world -- i.e. orders and inventory
• Some relationships are real but not important because we take no actions in the model and/or real world -- i.e. employee golf score and his salary
You will see other terms for these concepts in the literature, but a relationship has

- Degree
- Rank
- Membership
• Rank = number of participants in the relationship
  - Rank can be unary, binary or n-ary

• We like binary relationships
  - Write them as infixed operators in math
  - ER diagrams and graphs lead you to them

• A marriage is binary (husband, wife)

• A sale is ternary (buyer, seller, item)
  - The most important leg on a three-legged stool is the one that is missing
Degree

- Degree = number of elements from each entity class that take part in the relationship
- one to one = 1:1 = Conventional marriage
- one to many = 1:n = A harem
- many to many = m:n = Corporate marriage
- Some writers also include “non-relations”
  - zero to zero = prohibition; bestiality
  - zero to one = no relationship; Dilbert’s love life
Membership

- Membership = How the entity can participate in the relationship
- Mandatory = no entity, no relationship
- Optional = entity is not required
- Combinations are sometimes trickier than you would think - there can be conditional memberships
  - A prayer circle requires 13 Jews
  - A customer buys one or more items
- Employee must have a department; Department needs one or more employees to exist.
- Employee must have a department; Department can be empty
- Employee can be unassigned; Department needs one or more employees to exist
- Employee can be unassigned; Departments can be empty
It is hard to decide if something is an entity, a relationship or both.

A marriage license is relationship
- If you destroy the license, you are still married
- You must divorce or kill your wife!

A bearer bond is a relationship
- If you destroy the bond, you are out of luck - the token *is* the relationship

A relationship can have attributes of its own
- A home loan related buyer-seller-lender, but it has an amount of its own
Lots of notations for the diagram; Erwin, Chen, Howe et al

Several popular software packages to draw them.

Bad news about E-R tools
- Planar graph problem
- Only one solution generated
- No constraints created
- You get one model that might not be in 3NF
- Lots of different versions of E-R

This is why we teach ORM, but you need to be able to read E-R
E-R Diagrams -2
Chen’s notation

Entity A

relationship

Entity B
E-R Diagrams -3
Chen’s notation

zero

One

Many
Chen’s notation

- One and only one
- Zero or one
- One or more
ORM Models - 1

- Semantic modeling
  - NI AM
  - Object Role Modeling

- Basic idea:
  - Describe your system in simple, formal, declarative sentences to generate a database.
  - This captures a high level model of the system
  - ORM can generate SQL directly
Visio supports InfoModeler, an FORM tool designed by Terry Halpin.
There are diagrams, but nobody uses them after the first month -- they just get in the way.
You get more than one schema from the model
The models are 3NF to 5NF