

# Database Design

Supertypes and Subtypes

# Objectives

This lesson covers the following objectives:

- Define and give an example of a subtype
- Define and give an example of a supertype
- State the rules relating to entities and subtypes, and give examples of each
- Apply the rules of supertype and subtype by evaluating the accuracy of ER diagrams that represent them
- Apply the rules of supertype and subtype and include them in a diagram when appropriate

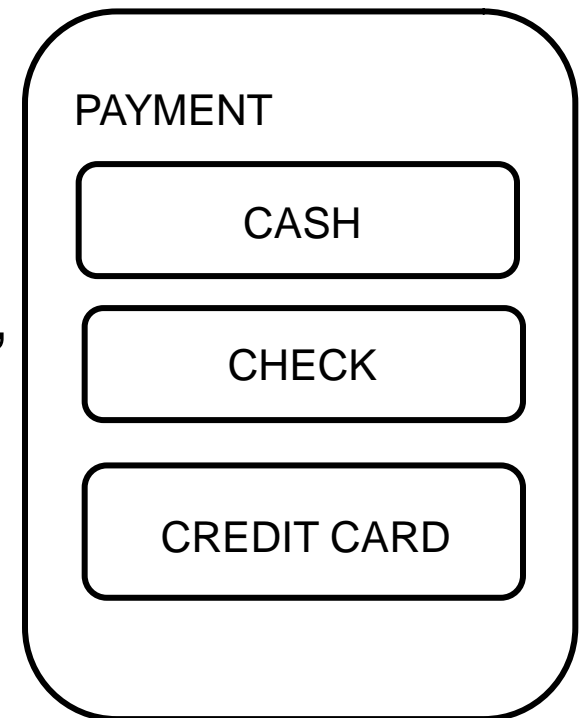
# Purpose

Supertypes and subtypes occur frequently in the real world -- food order types (eat in, to go), grocery bag types (paper, plastic), payment types (check, cash, credit). You can typically associate 'choices' of something with supertypes and subtypes. For example, what do you like on your sandwich? (mustard, mayonnaise, ketchup, onions, tomatoes, sprouts, cucumbers, lettuce, other).

Understanding real world examples helps us understand how and when to model them.

# Evaluating Entities

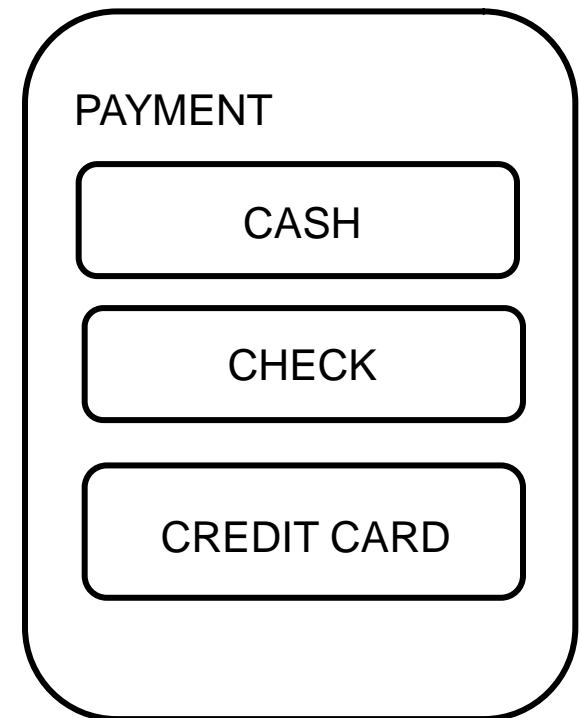
Often some instances of an entity have attributes and/or relationships which other instances do not have. Imagine a business which needs to track payments from customers. Customers can pay by cash, by check, or by credit card.



## Evaluating Entities (cont.)

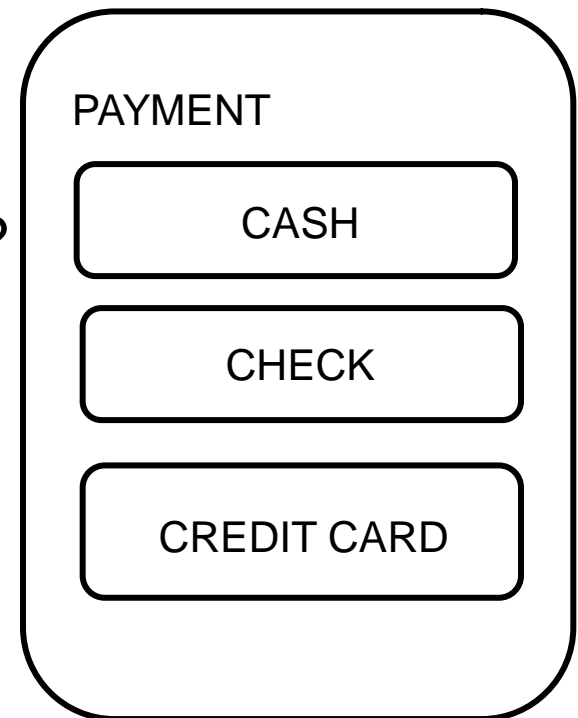
All payments have some common attributes: payment date, payment amount, and so on. But only credit cards would have a “card number” attribute.

And for credit card and check payments, we may need to know which CUSTOMER made the payment, while this is not needed for cash payments.



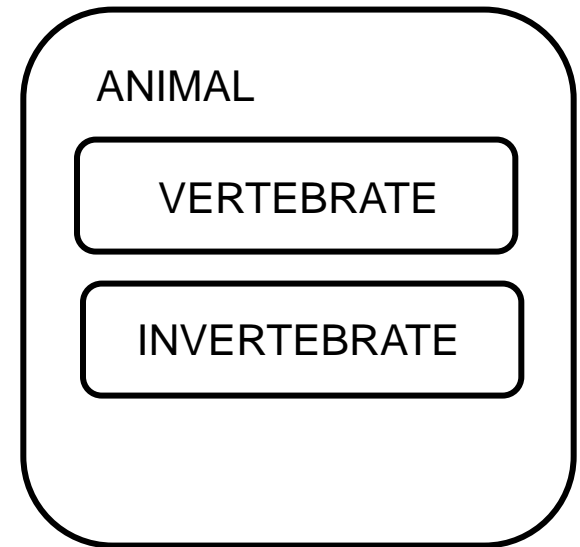
## Evaluating Entities (cont.)

Should we create a single PAYMENT entity or three separate entities CASH, CHECK, and CREDIT CARD? And what happens if in the future we introduce a fourth method of payment?



# Subdivide an Entity

Sometimes it makes sense to subdivide an entity into subtypes. This may be the case when a group of instances has special properties, such as attributes or relationships that exist only for that group. In this case, the entity is called a “supertype” and each group is called a “subtype”.

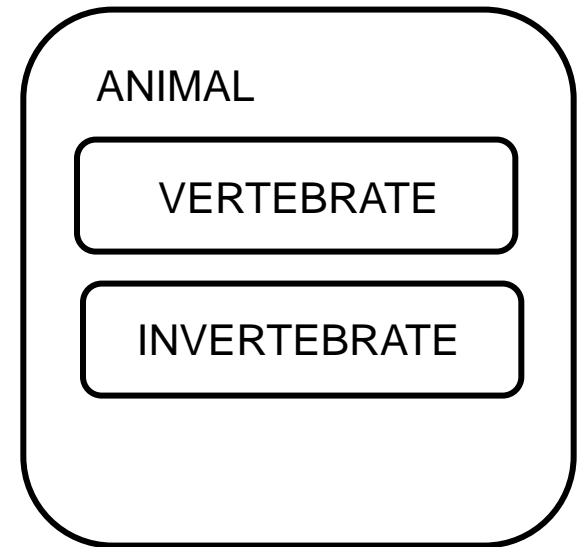


ANIMAL SUPERTYPE

## Subdivide an Entity (cont.)

A subtype:

- Inherits all attributes of the supertype
- Inherits all relationships of the supertype
- Usually has its own attributes or relationships
- Is drawn within the supertype
- Never exists alone
- May have subtypes of its own
- Is also known as a "subentity"

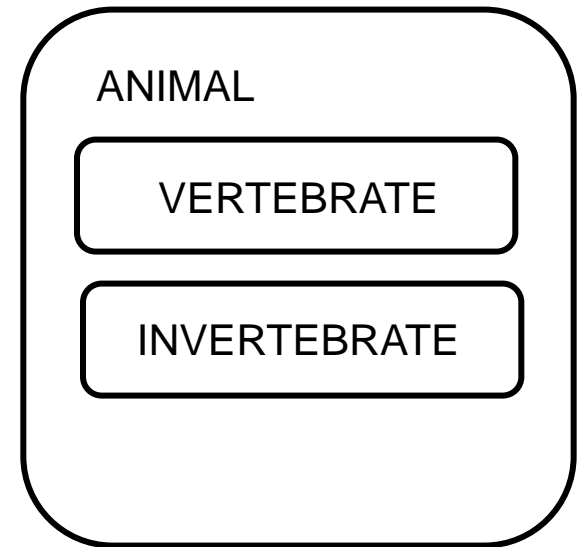


ANIMAL SUPERTYPE



## Subdivide an Entity (cont.)

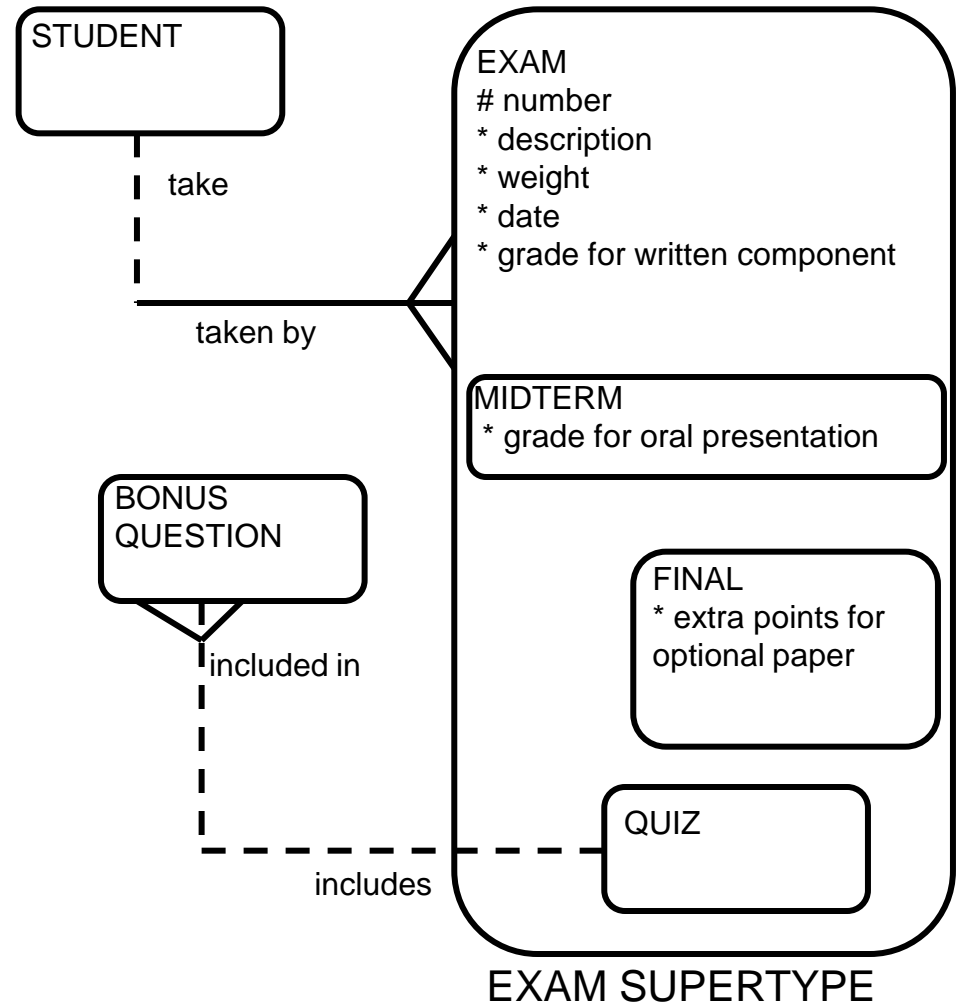
Invertebrates are animals with no backbond (such as earthworms).  
Vertebrates have backbone and can be subdivided into birds, mammals, reptiles, etc.



ANIMAL SUPERTYPE

# Supertype Example

EXAM is a supertype of QUIZ, MIDTERM, and FINAL. The subtypes have several attributes in common. These common attributes are listed at the supertype level. The same applies to relationships. Subtypes inherit all attributes and relationships of the supertype entity.



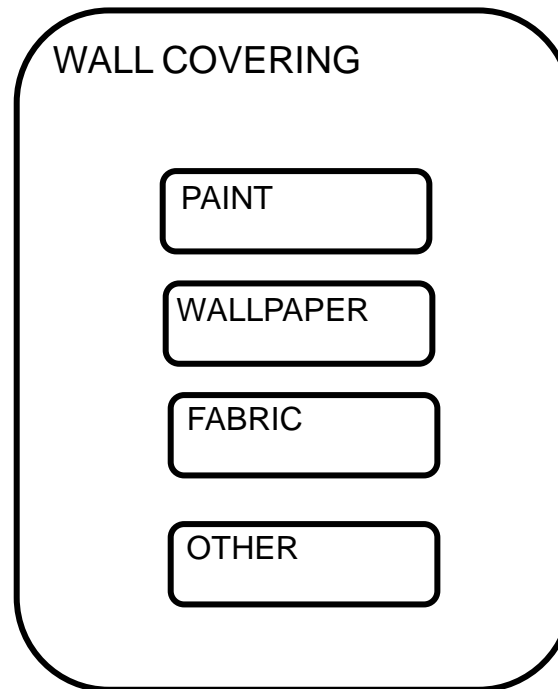
# Always More Than One Subtype

When an ER model is complete, subtypes never stand alone. In other words, if an entity has a subtype, a second subtype must also exist. This makes sense. A single subtype is exactly the same as the supertype. This idea leads to the two subtype rules:

**Exhaustive:** Every instance of the supertype is also an instance of one of the subtypes. All subtypes are listed without omission.

**Mutually Exclusive:** Each instance of a supertype is an instance of only one possible subtype.

# Always More Than One Subtype (cont.)



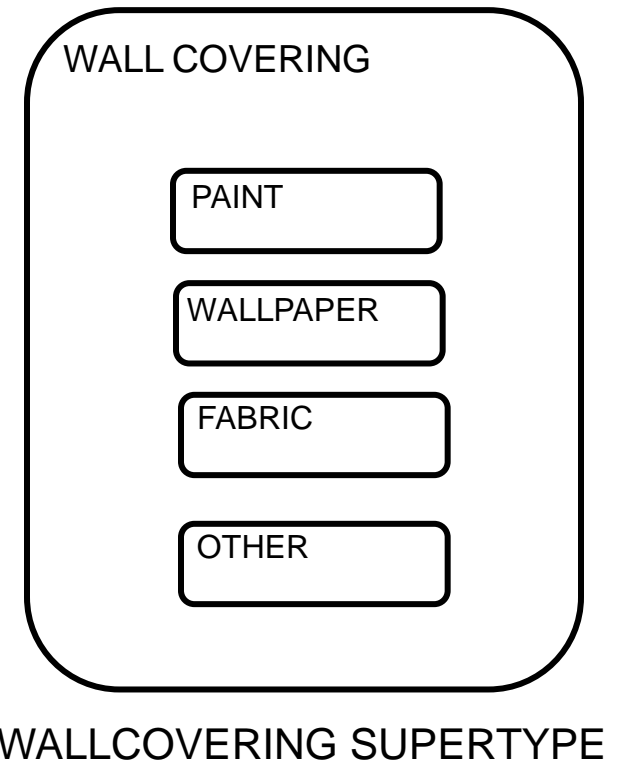
WALLCOVERING SUPERTYPE

At the conceptual modeling stage, it is good practice to include an **OTHER** subtype to make sure that your subtypes are exhaustive -- that you are handling every instance of the supertype.

# Subtypes Always Exist

Any entity can be subtyped by making up a rule that subdivides the instances into groups.

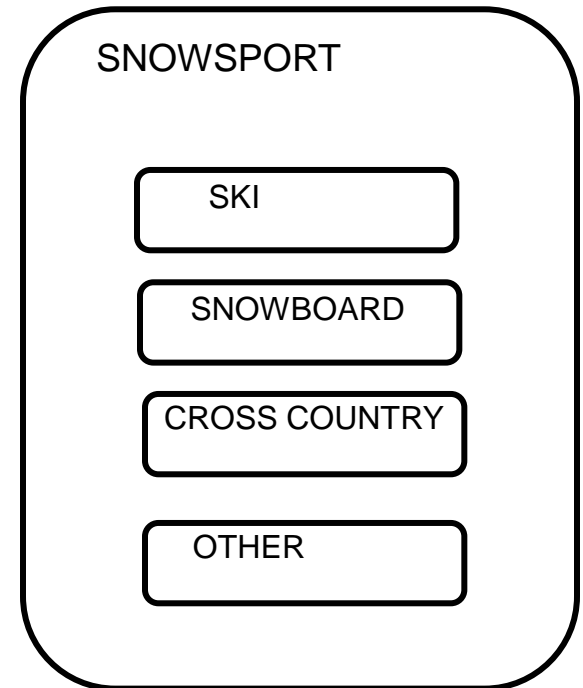
But being able to subtype is not the issue—having a reason to subtype is the issue. When a need exists within the business to show similarities and differences between instances, then subtype.



# Correctly Identifying Subtypes

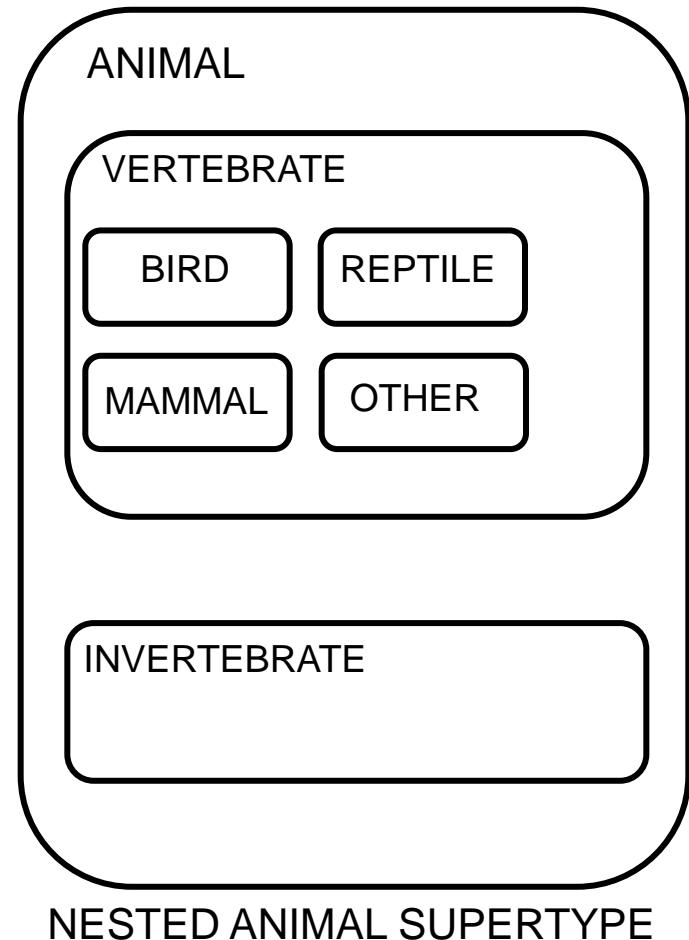
When modeling supertypes and subtypes, you can use three questions to see if the subtype is correctly identified:

1. Is this subtype a kind of supertype?
2. Have I covered all possible cases? (exhaustive)
3. Does each instance fit into one and only one subtype? (mutually exclusive)



# Nested Subtypes

You can nest subtypes. For ease of reading -- “readability” -  
- you would usually show subtypes with only two levels, but there is no rule that would stop you from going beyond two levels.



# Terminology

Key terms used in this lesson included:

- Exhaustive
- Mutually exclusive
- Subtype/subentity
- Supertype



# Summary

In this lesson, you should have learned how to:

- Define and give an example of a subtype
- Define and give an example of a supertype
- State the rules relating to entities and subtypes, and give examples of each
- Apply the rules of supertype and subtype by evaluating the accuracy of ER diagrams that represent them
- Apply the rules of supertype and subtype and include them in a diagram when appropriate