
ORACLE
Academy

# Database Design

## 7-2
## Hierarchies and Recursive Relationships

ORACLE Academy

# Objectives

This lesson covers the following objectives:

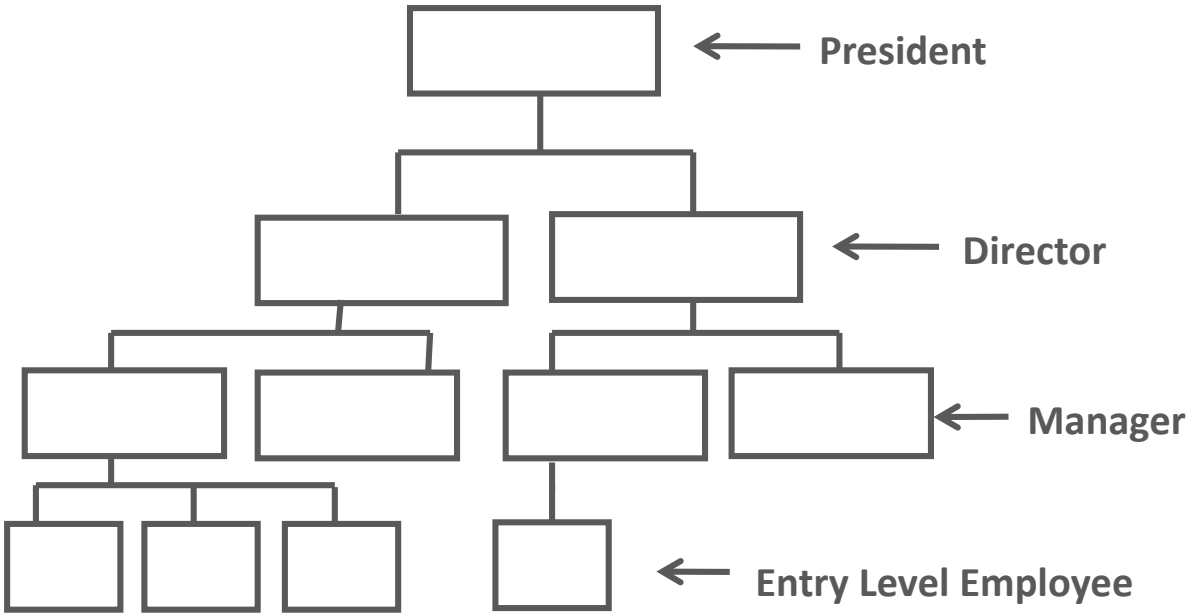
- Define and give an example of a hierarchical relationship
- Identify the UIDs in a hierarchical model
- Define and give an example of a recursive relationship
- Represent a recursive relationship in an ERD given a scenario
- Construct a model using both recursion and hierarchies to express the same conceptual meaning

# Purpose

- Often, roles are organized by hierarchy -- at work (manager, crew chief, front-counter clerk, food preparers), or in school (headmaster or principal, assistant headmaster or assistant principal, teachers, staff).
- Hierarchical data is very common.
- Understanding it will help you model:
  - Business organizational charts
  - Building structures
  - Family trees
  - and many other hierarchies found in the real world

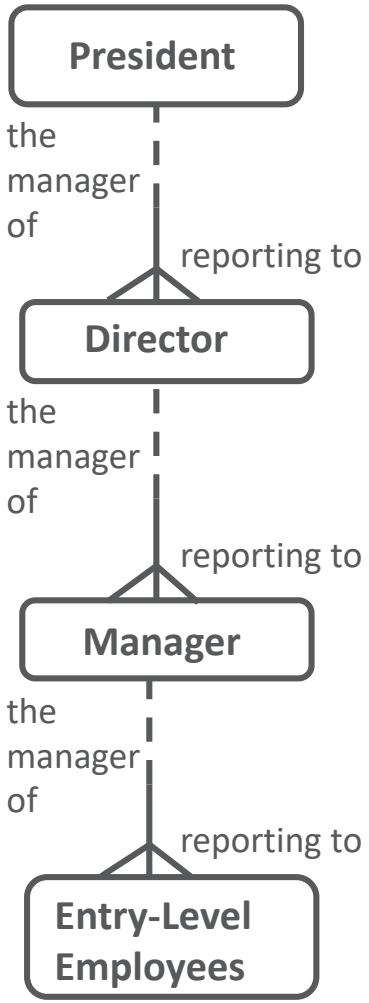
# Relationships in an Organizational Chart

- An Organization's reporting hierarchy can be represented by this organizational chart .



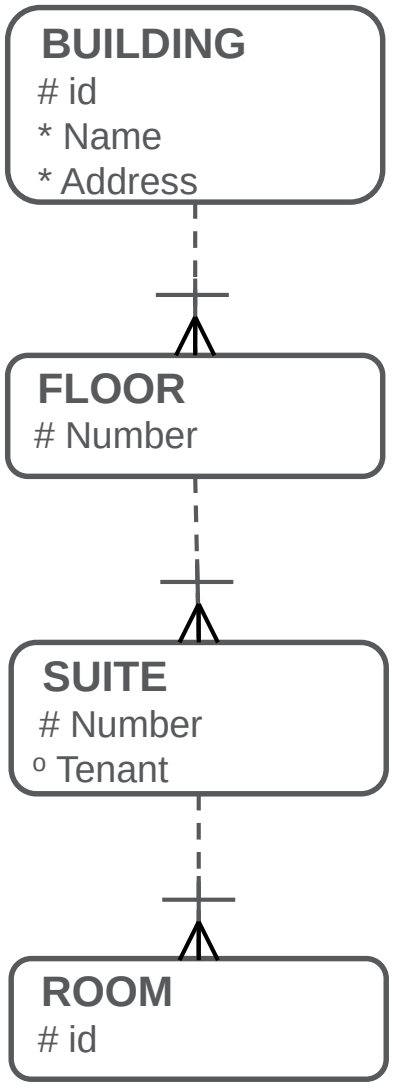
# Relationships in an Organizational Chart

- An organizational chart can be represented by this data model.
- We create an entity for each level, with a relationship to the next level.
- What are the UIDs for each entity?



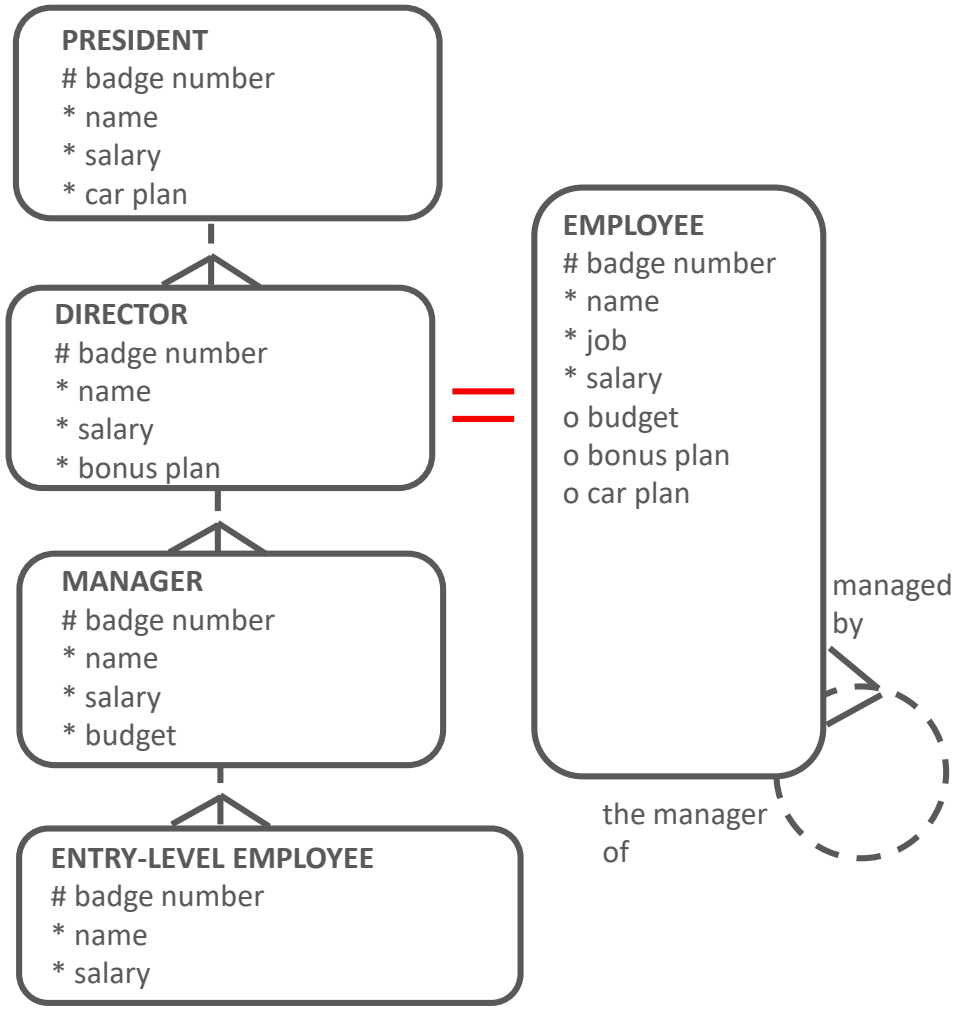
# Another Relationship Example

- Notice the barred relationships.
- Here you have a case of the cascading UIDs:
  - the UID of FLOOR is the combination of FLOOR number and the BUILDING id
  - the UID of SUITE is the combination of SUITE number and the FLOOR number and the BUILDING id
  - the UID of ROOM is the combination of ROOM id and SUITE number and FLOOR number and the BUILDING id



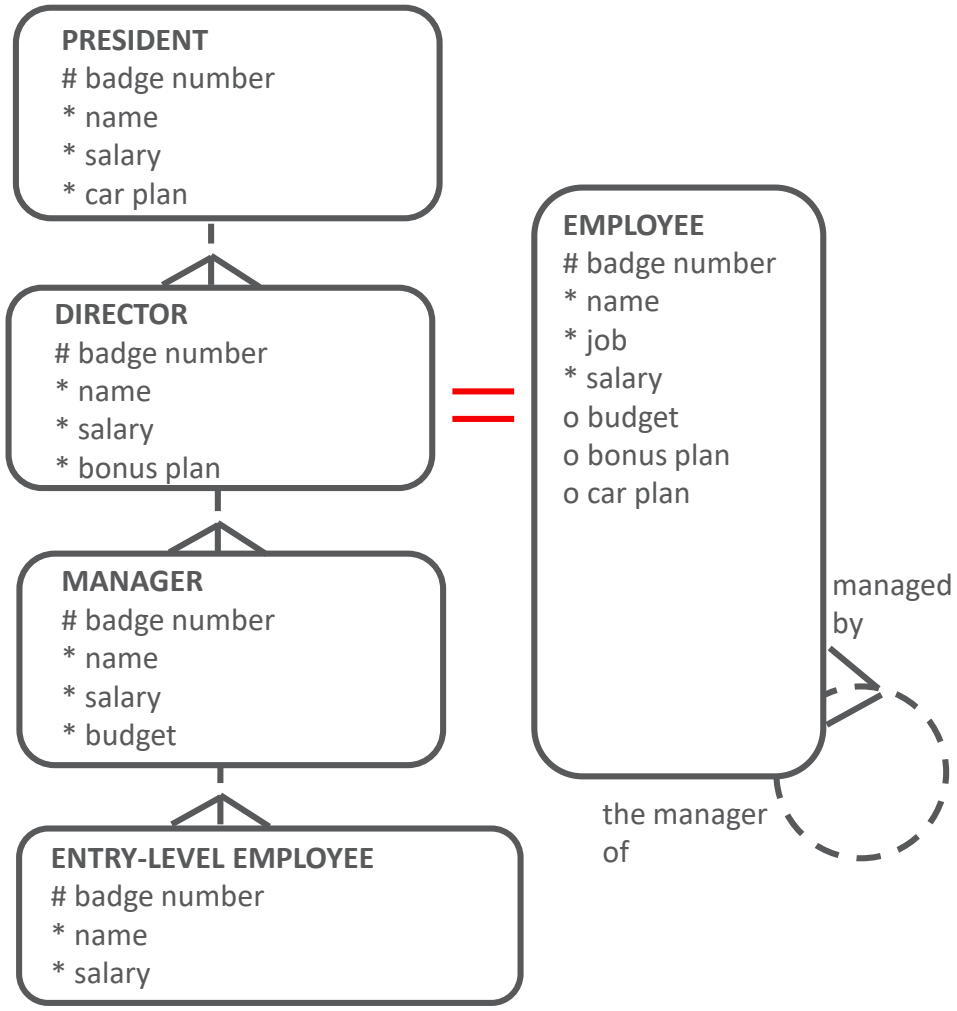
# Hierarchy Versus Recursive Relationship

- Both of these models represent all employees.
- The one on the left is a hierarchical structure.
- The one on the right uses a recursive relationship.



# Hierarchy Versus Recursive Relationship

- A relationship cannot be both hierarchical and recursive at the same time.
- Which one do you think is better?



# Hierarchy Versus Recursive Relationship

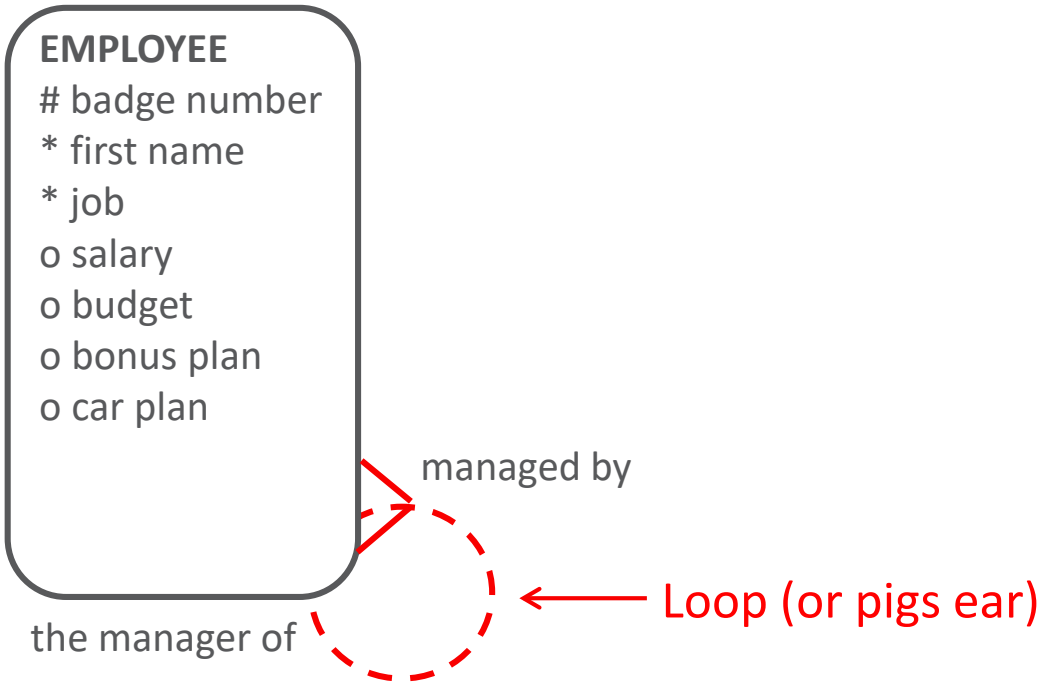
- Hierarchical: Hierarchical structures are more explicit and are easier for most people to understand because they are very similar to an organizational chart.
- Each entity can have its own mandatory attributes and relationships, if the business requires this (instead of all optional attributes and relationships, as you would have in a recursive).
- In this way, your data model truly reflects the business rules.

# Hierarchy Versus Recursive Relationship

- Recursive: Recursive relationships tend to be simpler because you are using only one entity.
- Your diagram will be less “busy.”
- However, they are less specific – you cannot have mandatory attributes or relationships unless they are mandatory in all instances of the entity.

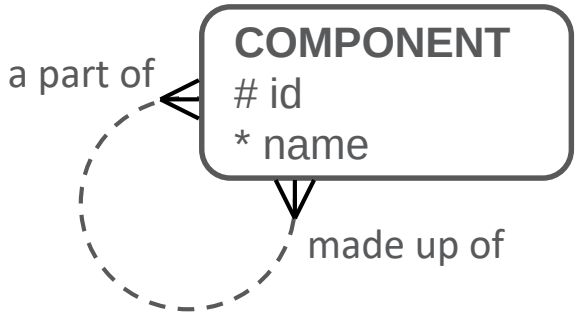
# Drawing Convention

- The ERD convention to show a recursive relationship is drawn as a loop, also known as a “pig’s ear”.



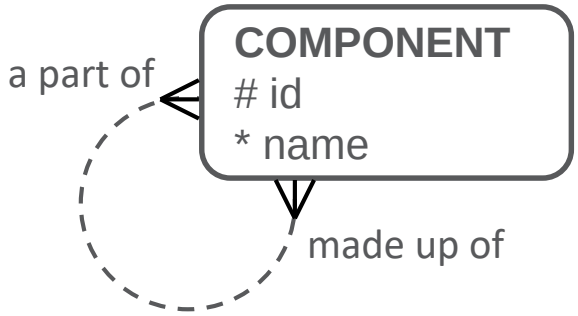
# Automobile Manufacturing Business Scenario

- For an automobile manufacturing organization, consider all elementary parts, subassemblies, assemblies, and products as instances of an entity called COMPONENT.
- The model can be created as a simple recursive relationship.



# Automobile Manufacturing Business Scenario

- Model Bill of Materials data as a many-to-many recursive relationship:
  - Each COMPONENT may be a part of one or more COMPONENTS.
  - Each COMPONENT may be made up of one or more COMPONENTS.



# Terminology

Key terms used in this lesson included:

- Hierarchal relationship
- Recursive relationship

# Summary

In this lesson, you should have learned how to:

- Define and give an example of a hierarchical relationship
- Identify the UIDs in a hierarchical model
- Define and give an example of a recursive relationship
- Represent a recursive relationship in an ERD given a scenario
- Construct a model using both recursion and hierarchies to express the same conceptual meaning


ORACLE
Academy