# The Layman's Guide to Reading Data Models

A data model shows a data asset's structure, including the relationships and constraints that determine how data will be stored and accessed.

1. Common Types of Data Models

Conceptual Data Model

A **conceptual data model** defines high-level relationships between real-world entities in a particular domain. Entities are typically depicted in boxes, while lines or arrows map the relationships between entities (as shown in Figure 1).


Figure 1: Conceptual Data Model
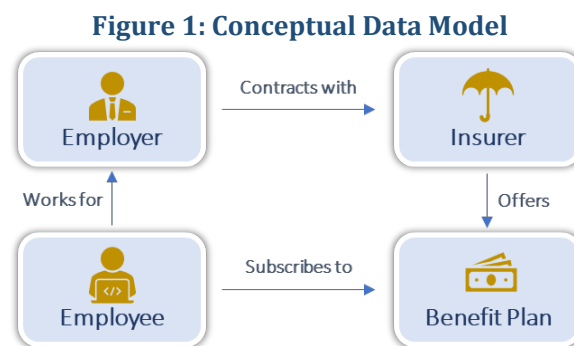
Logical Data Model

A **logical data model** defines how a data model should be implemented, with as much detail as possible, without regard for its physical implementation in a database. Within a logical data model, an entity's box contains a list of the entity's **attributes**.

One or more attributes is designated as a primary key, whose value uniquely specifies an instance of that entity. A primary key may be referred to in another entity as a **foreign key**.

In the Figure 2 example, each Employee works for only one Employer. Each Employer may have zero or more Employees. This is indicated via the model's line notation (refer to the Describing Relationships section).

Figure 2: Logical Data Model
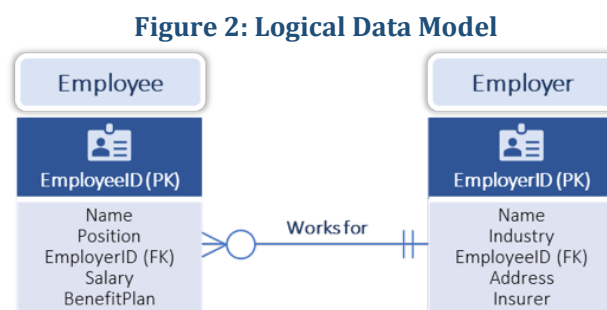
Physical Data Model

A physical data model describes the implementation of a data model in a database (as shown in Figure 3). Entities are described as tables, Attributes are translated to table column, and Each column's data type is specified.

Figure 3: Physical Data Model
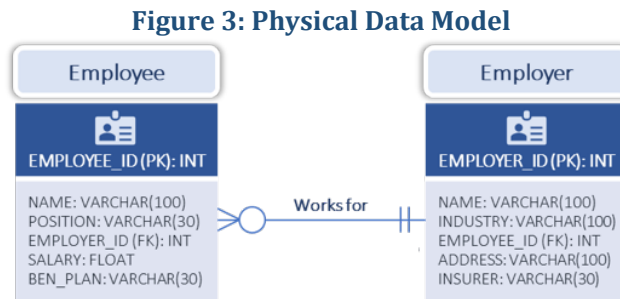
2. Describing Relationships

Ordinality and Cardinality

Logical and physical data models describe two entities' **ordinality** and **cardinality**, or the minimum and maximum number of times an instance of one entity can relate to instances of another entity.
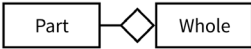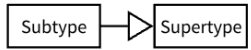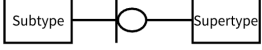
Line Notation Style

Different data models use different styles of line notation to indicate ordinality, cardinality, and other types of relationships between entities. In the examples above, ordinality and cardinality are described using crow's foot notation (the symbols at the end of each line).

Common notations in **Unified Modeling Language (UML)**, crow's foot, and **Integration DEFinition for Information Modeling (IDEF1X)** notation are described in the following table:

Table 1: Syntax in Common Data Modeling Notation Styles

Notation	Crow's Foot	UML	IDEF1X
One		N/A	N/A
Many		N/A	N/A
Zero or one		0..1	Z
One only		1	1
One or more		1..*	P
Zero or more		0..*	
(Specific range)	N/A	3..7	N/A
Composition*	N/A	Part ◆ Whole	"Is part of"

Notation	Crow's Foot	UML	IDEF1X
Aggregation*	N/A	Part ◇ Whole	"Is part of"
Subtype**	N/A	Subtype ▷ Supertype	Subtype ─○─ Supertype

***Aggregation** and **composition** are specific kinds of relationships. Aggregation means one entity can exist independently of another entity (i.e., an Employee and a Benefit Plan). Composition means one entity can't exist independently of another entity (i.e., an Employee must have an Employer).

A **subtype is an entity that has a parent-child relationship with another entity, a **supertype**. A supertype has attributes that are common to all of its subtypes.