# DATA MODELING

# Introduction

- Process of creating a data model for an information system by applying formal data exploration techniques.
- Process used to define and analyze data requirements needed to support the business processes.
- Therefore, the process of data modeling involves professional data modelers working closely with business stakeholders, as well as potential users of the information system.

# What is Data Model

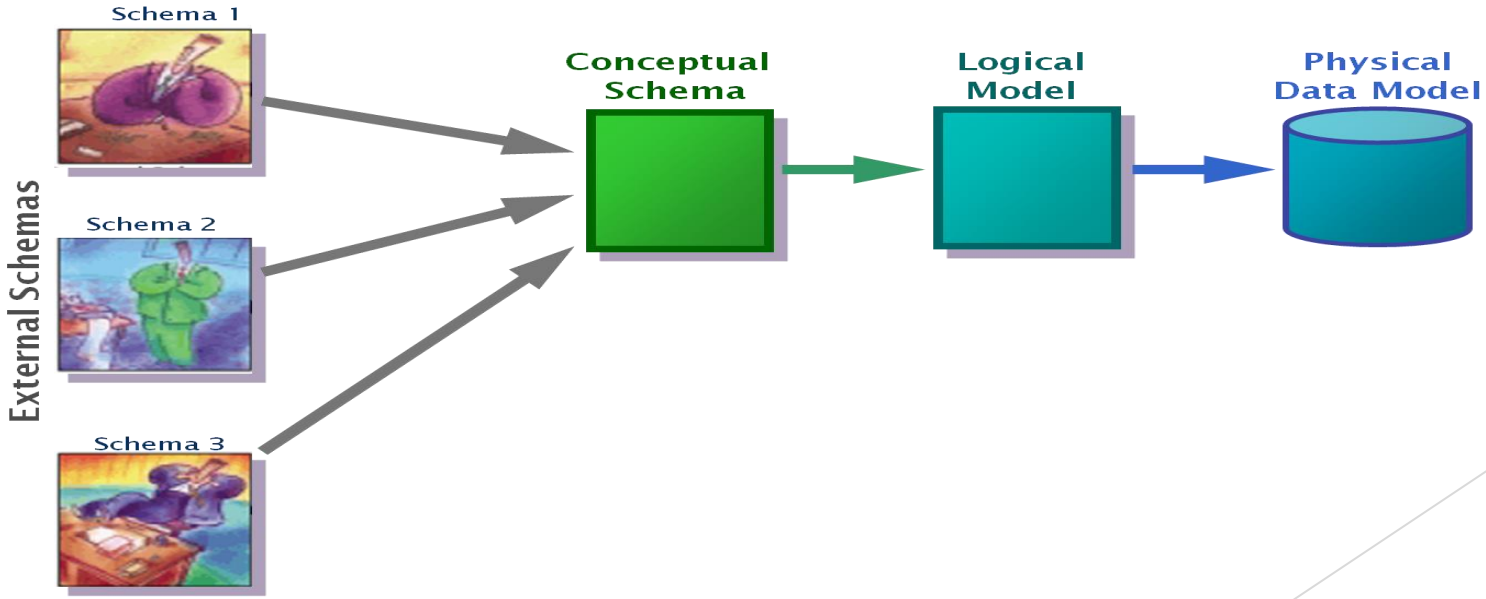
- Data Model is a collection of conceptual tools for describing data, data relationships, data semantics and consistency constraint.
- A data model is a conceptual representation of data structures required for data base and is very powerful in expressing and communicating the business requirements.
- A data model visually represents the nature of data, business rules governing the data, and how it will be organized in the database.

- A data model provides a way to describe the design of a database at the physical, logical and view levels.

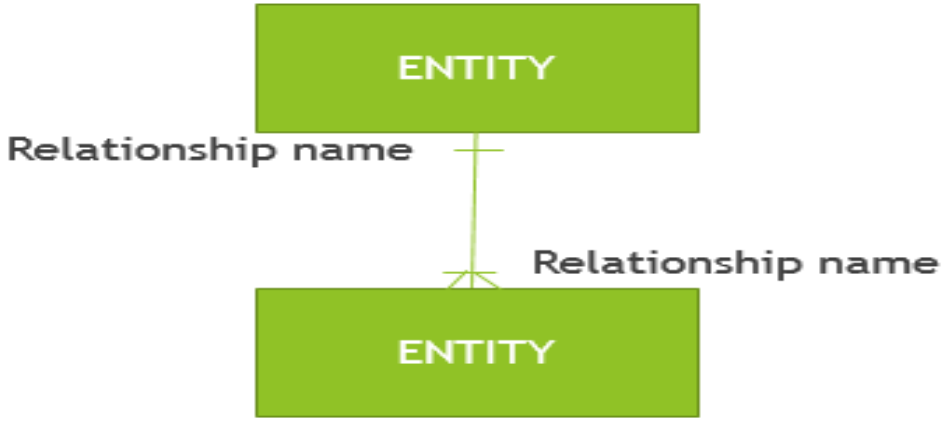
- There are three different types of data models produced while progressing from requirements to the actual database to be used for the information system

# Different Data Models

- Conceptual: describes WHAT the system contains.
- Logical: describes HOW the system will be implemented, regardless of the DBMS.
- Physical: describes HOW the system will be implemented using a specific DBMS.



A data model consists of entities related to each other on a diagram:



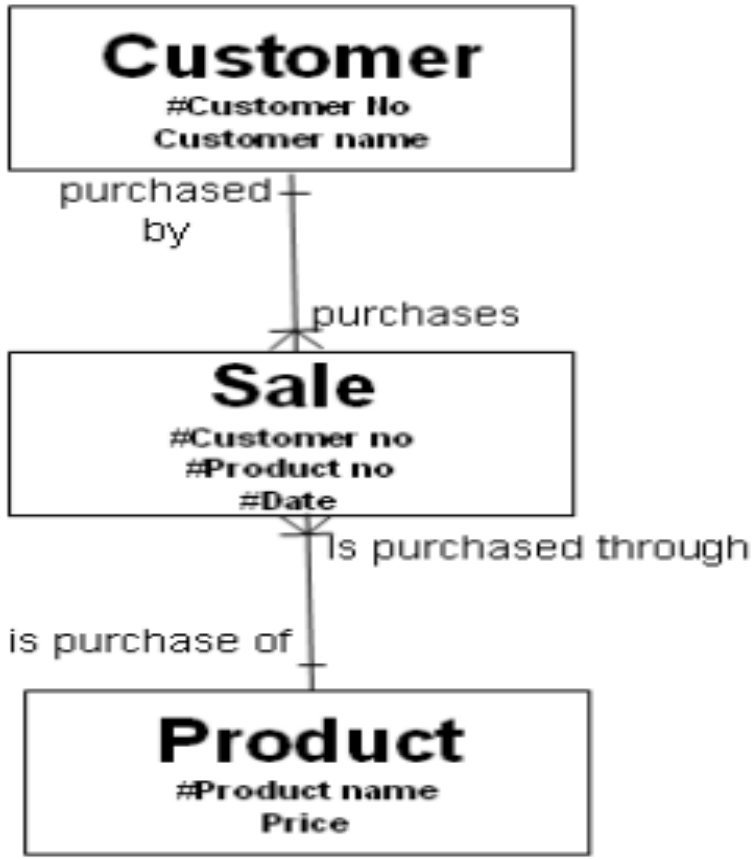

| Data Model Element | Definition |
|---|---|
| Entity | A real world thing or an interaction between 2 or more real world things. |
| Attribute | The atomic pieces of information that we need to know about entities. |
| Relationship | How entities depend on each other in terms of why the entities depend on each other (the relationship) and what that relationship is (the cardinality of the relationship). |

## Example:

Given that …

- “Customer” is an entity.
- “Product” is an entity.
- For a “Customer” we need to know their “customer number” attribute and “name” attribute.
- For a “Product” we need to know the “product name” attribute and “price” attribute.
- “Sale” is an entity that is used to record the interaction of “Customer” and “Product”.

Here is the diagram that encapsulates these rules:



# Notes

- By convention, entities are named in the singular.
- The attributes of “Customer” are “Customer No” (which is the unique identifier or primary key of the “Customer” entity and is shown by the # symbol) and “Customer Name”.
- “Sale” has a composite primary key made up of the primary key of “Customer”, the primary key of “Product” and the date of the sale.
- Think of entities as tables, think of attributes as columns on the table and think of instances as rows on that table:

Customer *(entity)*

| No *(attribute)* | Name *(attribute)* | |
|---|---|---|
| 10 | Fred Bloggs | *(instance)* |
| 67 | Freda Jones | *(instance)* |

Sale

| Customer No | Product Code | Date |
|---|---|---|
| 10 | 101 | 21/2/2020 |
| 67 | 452 | 22/2/2020 |

Product

| Code | Name | Price |
|---|---|---|
| 101 | Flange | £123.00 |
| 452 | Blitwort | £34.50 |

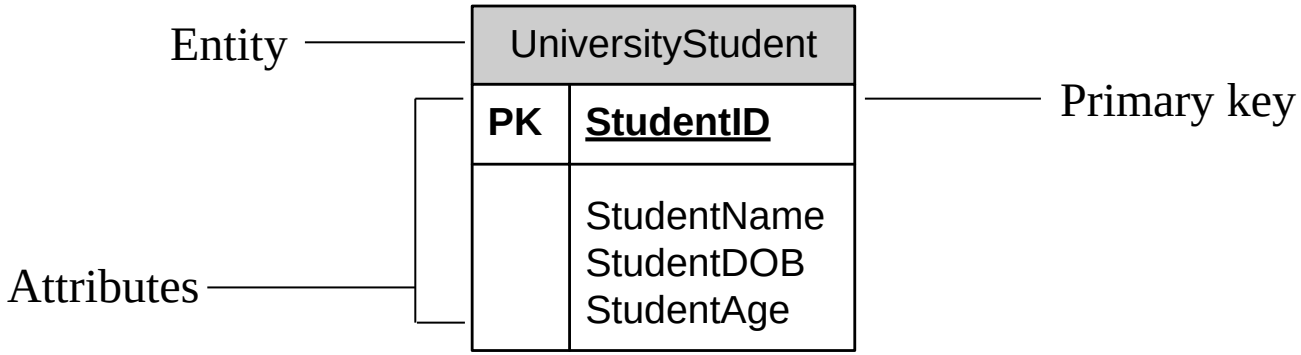
- If we want to know the price of a Sale, we can 'find' it by using the "Product Code" on the instance of "Sale" we are interested in and look up the corresponding "Price" on the "Product" entity with the matching "Product Code".

# Types of Data Models

- Entity-Relationship (E-R) Models
- UML (unified modeling language)

# Entity-Relationship Model

- Entity Relationship Diagrams (ERD) as this is the most widely used
- ERDs have an advantage in that they are capable of being normalized




- Represent entities as rectangles
- List attributes within the rectangle

- The ER model defines the three most relevant steps.
- It works around real-world entities and  the associations among them.
- At view level, the  ER model is considered a good option for designing databases.

- ❑ Requirement Analysis
- ❑ Conceptual Database Design
- ❑ Logical Database Design

# Requirement Analysis

- ✓ The very first step in designing a database application is to understand
  - ✓ what data is to be stored in the database,
  - ✓ what applications must be built on the top of it,
  - ✓ what operations are we must find out what the users want from the database.

# Conceptual Database Design

- The information gathered in the requirements analysis step is used to develop a high-level description of the data to be stored in database, along with the constraints known to holdover the data.
- The ER model is one of the high-level or semantic, data models used in database.

# Logical Database Design

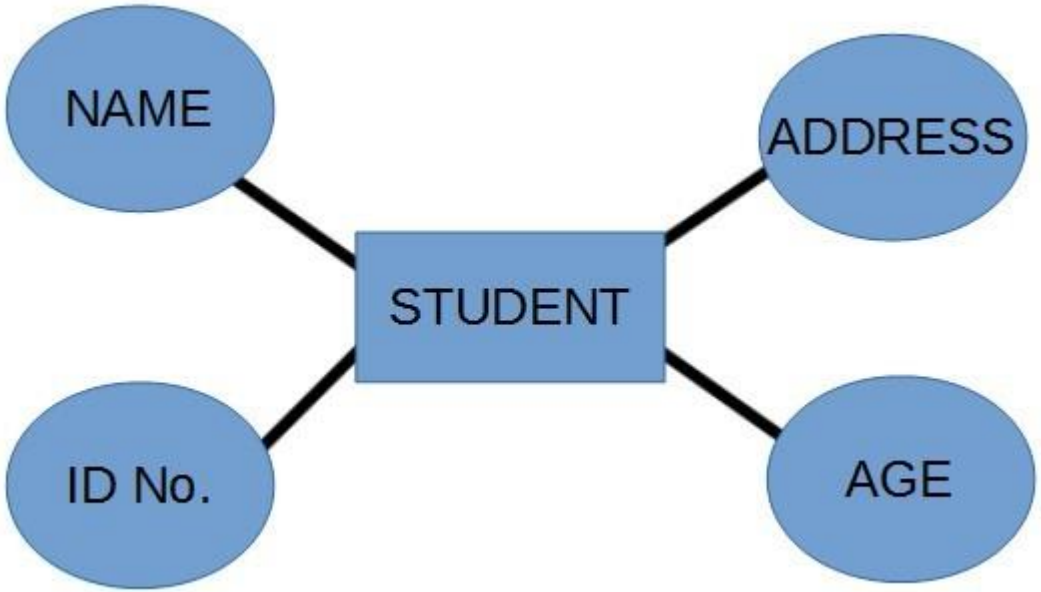
- We must choose a DBMS to implement our database design, and convert the conceptual database design into a database schema in the data model of chosen DBMS.
- Sometimes conceptual schema is called logical schema in Relational Data Model.

# Entity

- An *entity* can be a real-world object, either animate or inanimate, that can be easily identifiable.
- For example, in a school database, students, teachers, classes, and courses offered can be considered as entities. All these entities have some attributes or properties that give them their identity.
- An *entity set* is a collection of similar types of entities.
- An entity set may contain entities with attribute sharing similar values.
- For example, a Students set may contain all the students of a schoo l; likewise a Teachers set may contain the teachers of a school from all faculties.
- Entity sets need not be disjoint.

## Attributes

- Entities are represented by means of their properties, called attributes. All attributes have values. For example, a student entity may have name, class, and age as attributes.
- There exists a domain or range of values that can be assigned to attributes. For example ,a student's name cannot be a numeric value. It has to be alphabetic. A student's age cannot be negative, etc.



# TypesofAttributes

- Simple attribute− Simple attributes are atomic values, which cannot be divided further. For example, a student's phone number is an atomic value of 10 digits.
- Compositeatribute−Composite attributes are made of more than one simple atribute.
- For example, a student's complete name may have first_name and last_name.
- Derivedatribute−Derivedatributesaretheatributesthatdonotexistinthephysical database,buttheirvaluesarederivedfrom otheratributespresentinthedatabase.For example,average_salaryinadepartmentshouldnotbesaveddirectlyinthedatabase,insteadit canbederived.Foranotherexample,agecanbederivedfrom data_of_birth.
- Single-valueatribute−Single-valueatributescontainsinglevalue.Forexample− Social_Security_Number.
- Multivaluedatribute−Multivaluedatributesmaycontainmorethanonevalues.For example, apersoncanhavemorethanonephonenumber,email_ad res,etc.

# Entity-SetandKeys

Keyis an attribute or collection of attributes that uniquely identifies an entity among entity set. For example, the roll_number of a student makes him/her identifiable among students.

There are mainly three types of keys:

- SuperKey- A set of attributes (one or more) that collectively identifiesan entity in an entity set.
- CandidateKey- A minimal super key is called a candidate key. An entityset may have more than one candidate key.
- PrimaryKey- A primary key is one of the candidate keys chosen bythe database designer to uniquely identify the entity set.

# Relationship

The asociationamongentitiesiscaledarelationship.Forexample,anemploye works_at a department ,a student enrolls in a course. Here, Works_at and Enrolls are called relationships.


- RelationshipSet:Asetofrelationshipsofsimilartypeiscaledarelationshipset.Likeentities,a relationshipto canhaveattributes.Theseattributesarecaleddescriptiveattributes.
- Degree of Relationship: The number of participating entities in arelationship defines the degree of the relationship.

  Binary=degre 2

  Ternary =degree 3

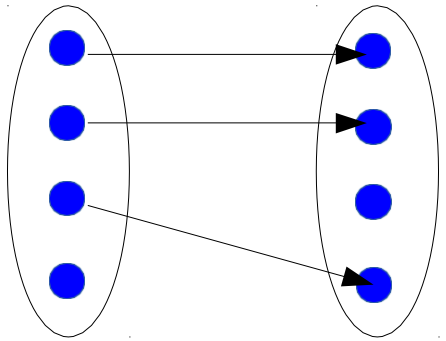
  n-ary =degree n

# MappingCardinalities

Cardinality defines the number of entities in one entity set,which can be associated with the number of entities of other set via relationship set.

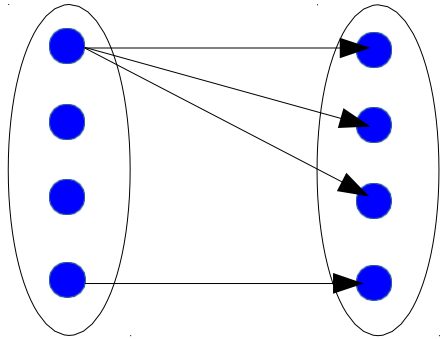
# 1.One-to-one

OneentityfromentitysetAcanbe asociatedwithatmostoneentity ofentitysetBandviceversa.



## 2.One-to-many

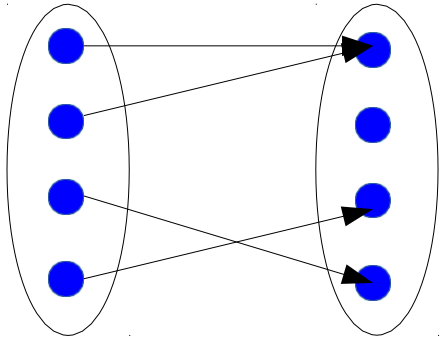
Oneentityfrom entitysetAcanbeasociatedwithmore thanoneentitiesofentitysetBhoweveranenti tyfrom entitysetB,canbeasociatedwithatmostonee ntity



# TypesofCardinalities(cont… . )

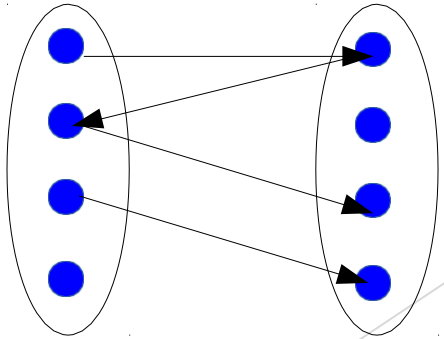
- ManytoOne

  More than one entities from entity setA can be associated with at most one entity of entity setB,howeveranentityfrom entity setBcanbeasociatedwithmorethanone entityfrom entitysetA.



- ManytoMany

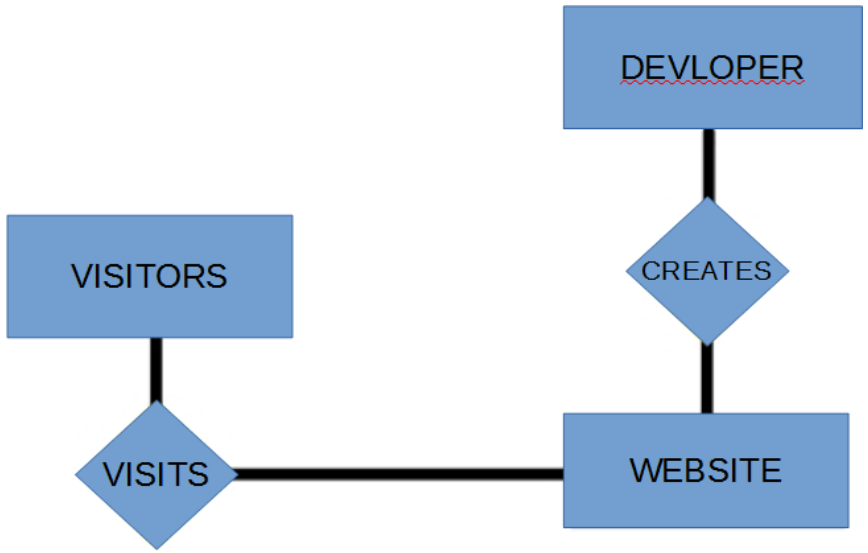
  One entity fromA can be asociated with more than one entity from B and viceversa.



# ENTITYRELATIONSHIPDIAGRAM

- Introduction
- Symbols & Notations
- Information Engine ring Style Used in ER-Diagram

# Introduction




ER-Diagramisavisual representationofdata thatdescribeshow dataisrelatedtoeach other.

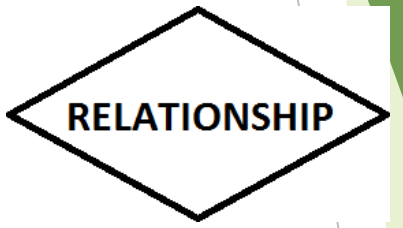
# SYMBOLSANDNOTATIONS

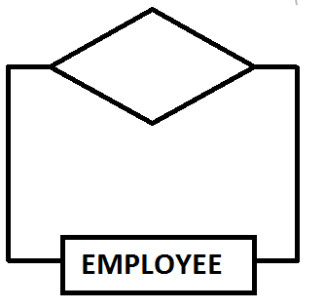
- Entities:Anentityisanobjector conceptaboutwhichyouwanttostoreinformation.


*Aweakentity*isanentitythatmust definedbyaforeignkeyrelationshipwith anotherentityasitcan otbeuniquelyidentifiedbyits ownattributesalone.

# SYMBOLSANDNOTATIONS(cont….)

- Actions:Actionsarerepresentedby diamondshape,andshowhowtwo, entitiesshareinformationinthedatabas



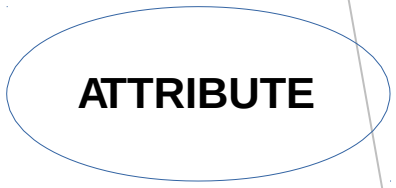




In some cases, entities can be Self-linked.

Forexample,employescansuperviseotheremployes

# SYMBOLSANDNOTATIONS(cont….)

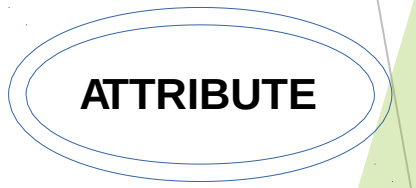
- Attributes: Attributes are represented by ovals. A key attribute is the unique, distinguishing characteristic of the entity.




For example, an employee's social security number might be the employe's key atribute.

A multivalued attribute can have more than one value.

For example, an employee entity can have multiple skill values.



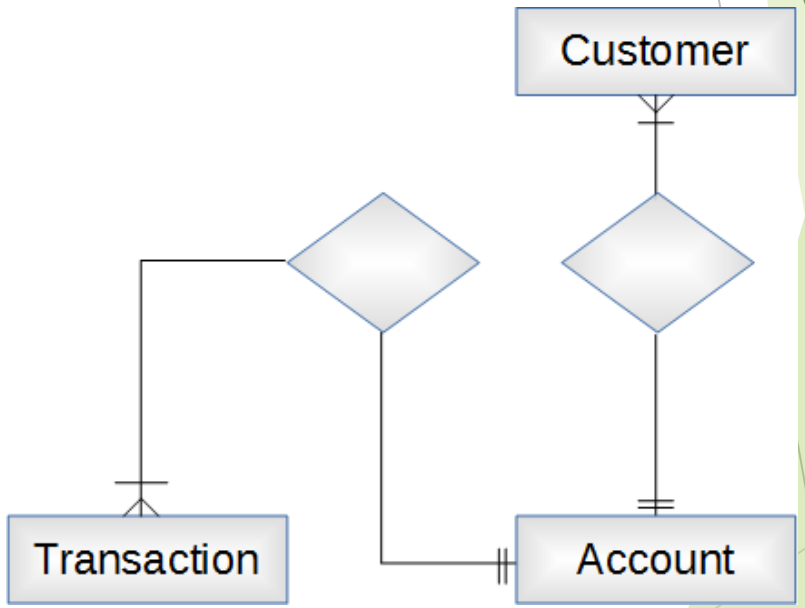

Aderivedatributeisbasedonanotheratribute.

Forexample,anemploye'smonthlysalaryisbasedonthe employe'sannualsalary.

# SYMBOLSANDNOTATIONS(cont….)

- Connectinglines:Solidlinesthatconnectatributesto showtherelationshipsofentitiesinthediagram.

- Cardinality:Cardinalityspecifieshowmanyinstances ofanentityrelatetooneinstanceofanotherentity. Ordinalityisalsocloselylinkedtocardinality.While cardinalityspecifiestheocurencesofarelationship, ordinalitydescribestherelationshipaseither mandatoryoroptional.Inotherwords,cardinality specifiesthemaximum numberofrelationshipsand ordinalityspecifiestheabsoluteminimum numberof relationships.



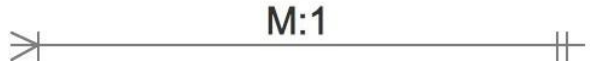
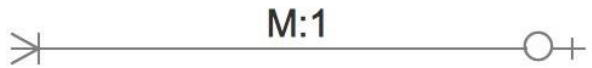
# INFORMATIONENGINEERINGS TYLE USEDINER-DIAGRAM

**Relationships (Cardinality and Modality)**

| SYMBOLS | MEANING |
| --- | --- |
|  | ZeroorMore |
|  | OneorMore |
|  | One and only One |
|  | ZeroorOne |

# INFORMATIONENGINEERINGSTYLE USEDINER-DIAGRAM(cont.)

**Many - to - One**

| SYMBOLS | MEANING |
| --- | --- |
|  | • Aonethroughmanynotationononeside ofarelationshipandaoneand onlyoneontheother. |
|  | • A zero through many notation on one sideofarelationshipandaoneand onlyoneontheother. |
|  | • Aonethroughmanynotationononeside of a relationship and a zero or one notationontheother. |
|  | • Azerothroughmanynotationononeside of a relationship and a zero or one notationontheother. |

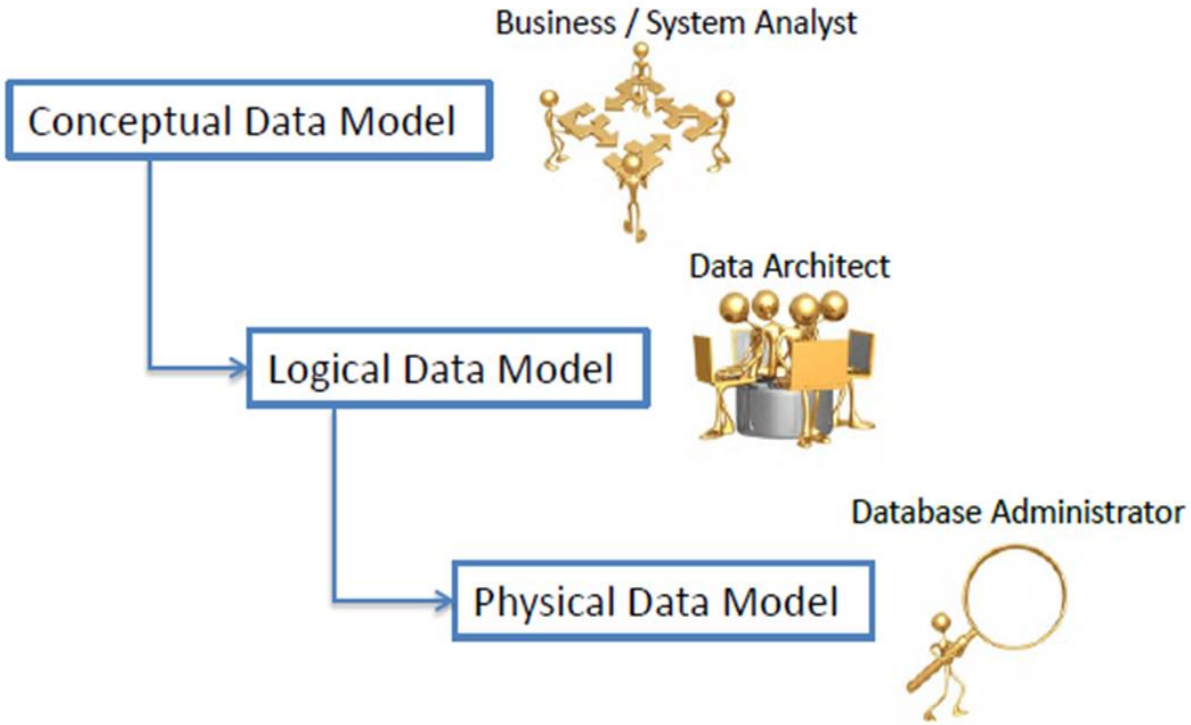
# INFORMATIONENGINEERINGSTYLE USEDINER-DIAGRAM(cont.)

**Many - to - One**

| SYMBOLS | MEANING |
| --- | --- |
|  | • A zero through many on both sides of a relationship. |
|  | • A one through many on both sides of a relationship. |
|  | • A zero through many on one side and a onethroughmanyontheother. |
|  | • Aoneandonlyonenotationononeside ofarelationshipandazerooroneon theother. |
|  | • Aoneandonlyonenotationonboth sides |

# Why and When

- The purpose of a data model is to describe the concepts relevant to a domain, the relationships between those concepts, and information associated with them.



- Used to model data in a standard, consistent, predictable manner in order to manage it as a resource.
- To have a clear picture of the base data that your business needs.
- To identify missing and redundant base data.

- To Establish a baseline for communication across functional boundaries within your organization.
- Provides a basis for defining business rules.
- Makes it cheaper, easier, and faster to upgrade your IT solutions.


THANK
YOU