Data dictionary store the following information.

(1) Name of data item

(2) Aliases

(3) Description

(4) Related Data Item

(5) Range of Values

(6) Data Structure Definition

The name of data item is self explanatory. Aliases include other names by which data item is called. Description is a textual description of what the data item is used for. Related data items capture relationship between data items. Range of values means that all the possible values *i.e.*, positive, negative, lower bounds, upper bounds etc. Data structure definition defines the composition of data items.

The following symbols are used for data dictionary.

| Notation | Name | Meaning |
|---|---|---|
| = | Definition or Composition | "Consist of" or and |
| + | Concentration | "and" |
| [ ] | Iteration | Multiple occurrence of |
| [1] | Selection | choice 1 or choice 2 |
| \| | Separator | separates alternatives |
| ** | Comments | additional information |
| ( ) | Options | may or may not be present. |
| @ | Identifier | keyfield for a store |

## Types of Data Dictionary

There are 2 types of data dictionaries.

(a) Integrated

(b) Stand Alone

An integrated dictionary is related to one database management system. To the extent the organisation data is under this DBMS it is global or organisation wide. However, very few enterprises have all their data eggs in one basket, so the dictionary documentation (metadata) can be considered as local and fragments.

The stand alone dictionary is not tied to any one DBMS, although it may have special advantages for DBMS, such as the IBM DB-DC Data Dictionary, which has special features related to the IBM IMS DBMS but is still a stand alone variety of dictionary.

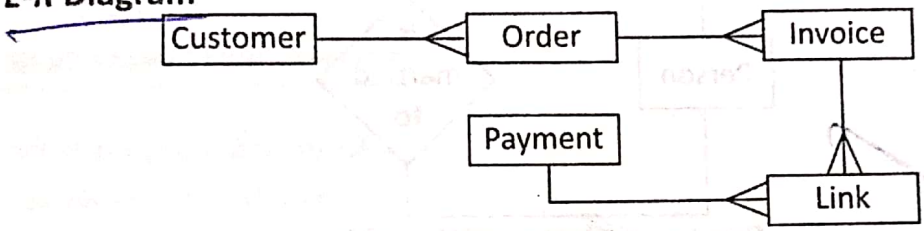
## ✔ Entry Relationship Diagram

An entity relationship model is an abstract conceptual representation of structured data. ER modeling is a relational scheme database modeling method, used to produce a type of conceptual data model of a system, often a relational database, and its requirements in a top-down fashion. Diagram created using the process is called entity relationship diagrams or ER diagrams.

ER diagram is detailed logical representation of data for an organization and uses three main components *i.e.*, data entities, their relationships and their associated attributes.

ER diagram is a data oriented model while DFD is a process oriented model. An ER diagram illustrates "data at rest" while DFD shows that in motion. An ERD is a more detailed picture of data stores, independent of the processes performed with those data stores. The ERD does not provide any information about the function that create or use that data. It focuses on the data relationships of the system. The ERD is beneficial as it highlights relationships between data stores of the DFD. Based on the information presented by the ERD, one is able to see what kind of keys or indices will need to access database records efficiently.

### Examples of *E-R* Diagram




ER diagram is a technique that is widely used in the world of business and information technology to show how information is, or should be, stored and used within a business system.

The success of any organization rules on the efficient flow and processing of information. In the above mentioned examples, information flows around the various departments with in the organization. This information can take many forms, for example, it could be written, oral or electronic.

### Components of *E-R* Diagram

There are 3 main components of ERD :

**1. Entity :** An entity is a fundamental thing of an organization about which data may be maintained. An entity has its own identity, which distinguishes it from each other entity. An entity is represented by a rectangular box or an ERD. We use capital letters in naming an entity types and in an E-R diagram the name is placed inside a rectangle *e.g.*,

INVOICE

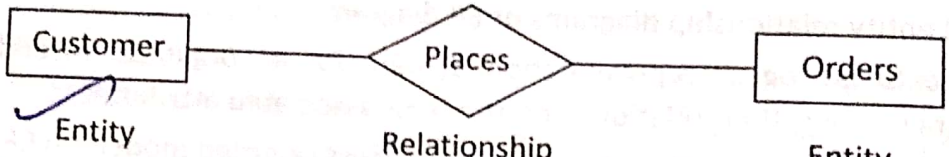
Entity represents a collection or set of objects and each object can be identified uniquely in some fashion *e.g.*, we have taken entity invoice and one should be able to distinguish one invoice from other. Each object can be described by one or more elements. *e.g.*, invoice object has various elements such as invoice number, data, quantity, discount, total price etc.

INVOICE CUSTOMER

Entity Types

**2. Relationship :** Objects are connected to one another by relationships or we can say a relationship is a reason for associating two entity types. Relationships are represented by diamond notation in the *ER* diagram.

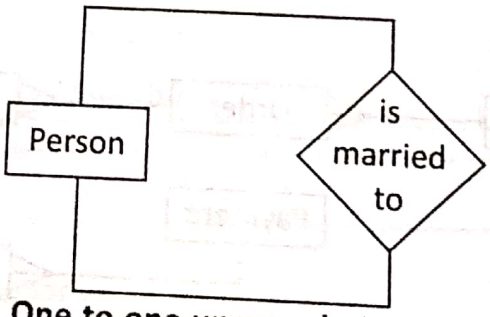
Relationships represents something that must be remembered by the system, something that is not calculated. Relationship is important to maintain database.




The number of entity types that participates in the relationships, is called the degree of relationship, *e.g.*, customer places orders.

The above mentioned relationship has degree two because it involves 2 entity types *i.e.*, customer and orders.

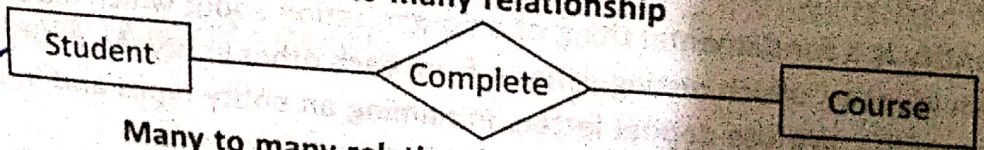
The three most common relationships in *E-R* diagram are unary *i.e.*, degree one, binary *i.e.*, degree 2, and ternary *i.e.*, degree 3.




**One to one unary relationship**


**One to one relationship**

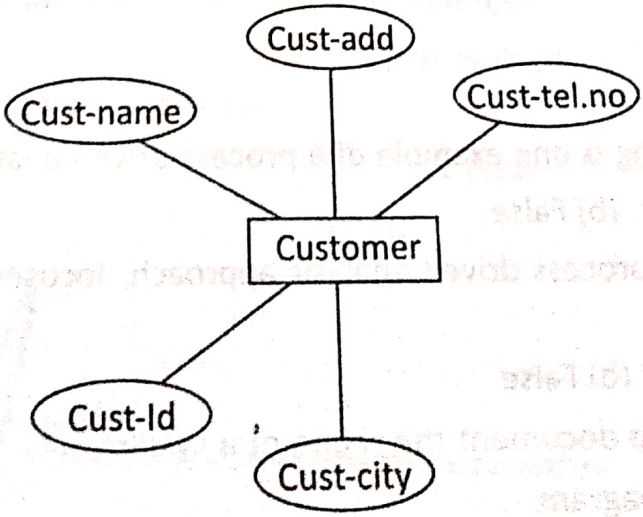

**One to many relationship**




**Many to many relationship Binary Relationships**

**3. Attributes :** Each entity type has set of attributes associated with it. An attribute is a property or characteristics of an entity that is of interest to the organization. In an *E-R* diagram, we can visually represent an attribute by placing its name as an ellipse with a line connecting it to the associated entity. Attribute is denoted by.


*e.g.,* CUSTOMER entity type can be represented by the following diagram.




## Exercises