# Chapter 4

# Entity-Relationship Diagram (ERD)

## 4.1 Introduction

An entity-relationship diagram (ERD) is a data modeling technique that graphically illustrates an information system’s entities and the relationships between those entities. An ERD is a conceptual and representational model of data used to represent the entity framework infrastructure.

Steps involved in creating an ERD include:

1. Identifying and defining the entities
2. Determining all interactions between the entities
3. Analyzing the nature of interactions/determining the cardinality of the relationships
4. Creating the ERD

An entity-relationship diagram (ERD) is crucial to creating a good database design. It is used as a high-level logical data model, which is useful in developing a conceptual design for databases.

An entity is a real-world item or concept that exists on its own. Entities are equivalent to database tables in a relational database, with each row of the table representing an instance of that entity.

An attribute of an entity is a particular property that describes the entity. A relationship is the association that describes the interaction between entities. Cardinality, in the context of ERD, is the number of instances of one entity that can, or must, be associated with each instance of another entity. In general, there may be one-to-one, one-to-many, or many-to-many relationships.

For example, let us consider two real-world entities, an employee and his department. An employee has attributes such as an employee number, name, department number, etc. Similarly, department number and name can be defined as attributes of a department. A department can interact with many employees, but an employee can belong to only one department, hence there can be a one-to-many relationship, defined between department and employee.

### 4.2 Components of E-R Diagram

**Entity relational diagram (ER Diagram)** is used to represent the requirement analysis at the conceptual design stage. the database is designed from the ER Diagram or we can say that ER Diagram is converted to the database.

Each entity in the ER Diagram corresponds to a table in the database.

The attributes of any an entity correspond to field of a table.

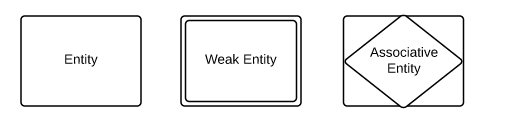
The ER Diagram is converted to the database.

The elements of an ERD are:

**1. ENTITIES**

Entities are objects or concepts that represent important data. They are typically nouns, e.g. *customer*, *supervisor*, *location*, or *promotion*.

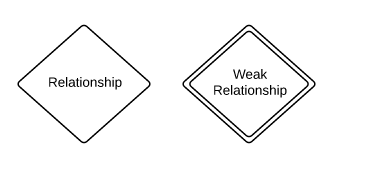
* **Strong entities** exist independently from other entity types. They always possess one or more attributes that uniquely distinguish each occurrence of the entity.
* **Weak entities** depend on some other entity type. They don't possess unique attributes (also known as a primary key) and have no meaning in the diagram without depending on another entity. This other entity is known as the owner.
* **Associative entities** are entities that associate the instances of one or more entity types. They also contain attributes that are unique to the relationship between those entity instances.



**2. RELATIONSHIPS**

* **Relationships** are meaningful associations between or among entities. They are usually verbs, e.g. *assign*, *associate*, or *track*. A relationship provides useful information that could not be discerned with just the entity types.
* **Weak relationships**, or identifying relationships, are connections that exist between a weak entity type and its owner.

#### Ternary Relationship, Relationship of degree three.



**3. ATTRIBUTES**

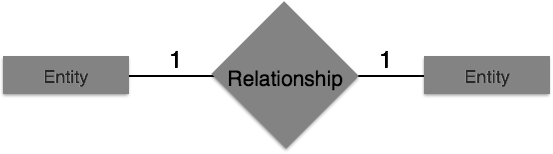
* **Attributes** are characteristics of either an entity, a many-to-many relationship, or a one-to-one relationship.
* **Multivalued attributes** are those that are capable of taking on more than one value.
* **Derived attributes** are attributes whose value can be calculated from related attribute values.
* **Composite** attributes  are represented by ellipses that are connected with an ellipse. they are further divided in a tree like structure. Every node is then connected to its attribute
* **Key** attribute represents the main characteristic of an Entity. It is used to represent Primary key. Ellipse with underlying lines represent Key Attribute.



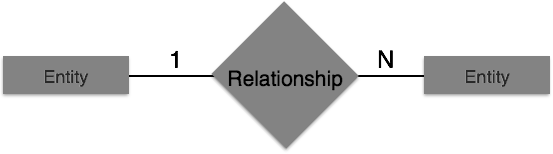
### 4.3 Binary Relationship and Cardinality

Cardinality is the number of instance of an entity from a relation that can be associated with the relation.

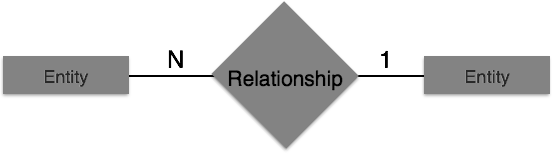
* **One-to-one** − When only one instance of an entity is associated with the relationship, it is marked as '1:1'. The following image reflects that only one instance of each entity should be associated with the relationship. It depicts one-to-one relationship.



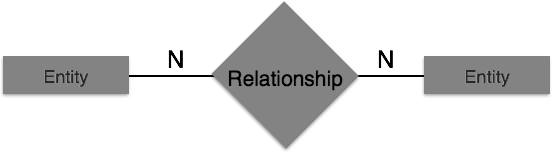
* **One-to-many** − When more than one instance of an entity is associated with a relationship, it is marked as '1:N'. The following image reflects that only one instance of entity on the left and more than one instance of an entity on the right can be associated with the relationship. It depicts one-to-many relationship.



* **Many-to-one** − When more than one instance of entity is associated with the relationship, it is marked as 'N:1'. The following image reflects that more than one instance of an entity on the left and only one instance of an entity on the right can be associated with the relationship. It depicts many-to-one relationship.



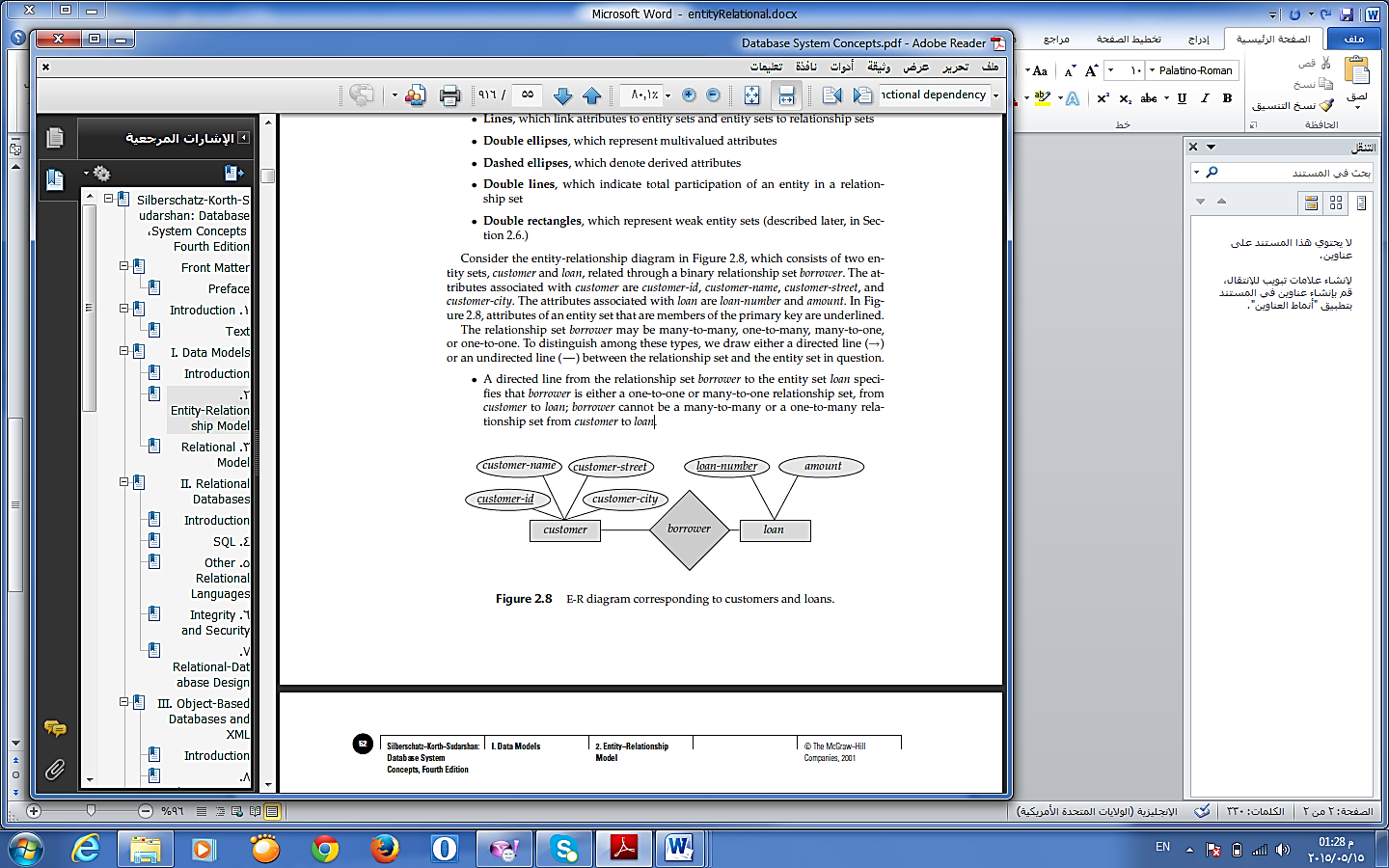
* **Many-to-many** − The following image reflects that more than one instance of an entity on the left and more than one instance of an entity on the right can be associated with the relationship. It depicts many-to-many relationship.



Consider the entity-relationship diagram in Figure 4.1, which consists of two entity sets, *customer* and *loan*, related through a binary relationship set *borrower*. The attributes associated with *customer* are *customer-id*, *customer-name*, *customer-street*, and *customer-city*. The attributes associated with *loan* are *loan-number* and *amount*. In Figure

4.1, attributes of an entity set that are members of the primary key are underlined.

The relationship set *borrower* may be many-to-many, one-to-many, many-to-one, or one-to-one. To distinguish among these types, we draw either a directed line (*→*)or an undirected line (—) between the relationship set and the entity set in question.

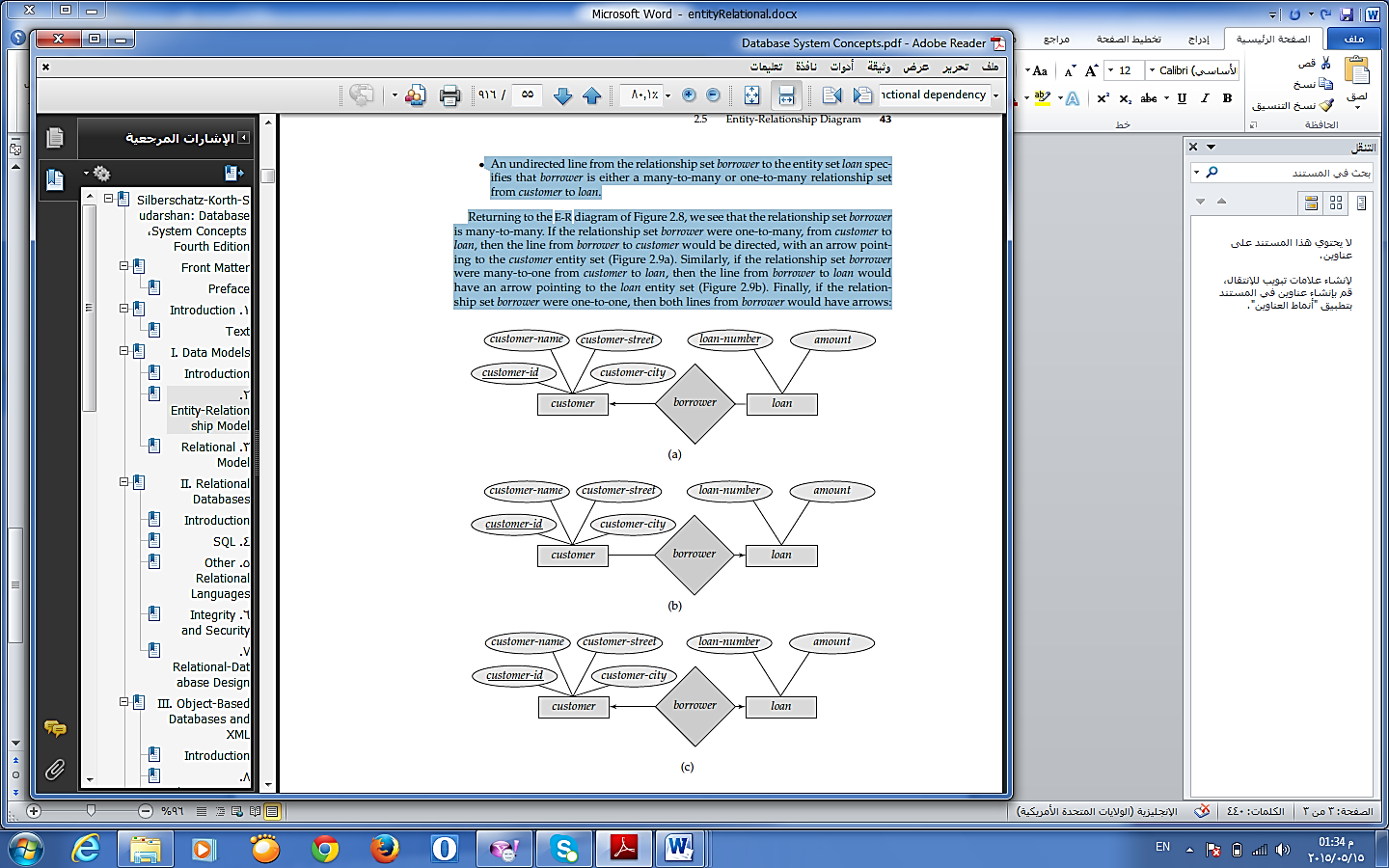
*•* A directed line (*→*) from the relationship set *borrower* to the entity set *loan* specifies that *borrower* is either a one-to-one or many-to-one relationship set, from *customer* to *loan*; *borrower* cannot be a many-to-many or a one-to-many relationship set from *customer* to *loan*.

**Figure 4.1 E-R diagram corresponding to customers and loans.**

* An undirected line (—) from the relationship set *borrower* to the entity set *loan* specifies that *borrower* is either a many-to-many or one-to-many relationship set from *customer* to *loan*.

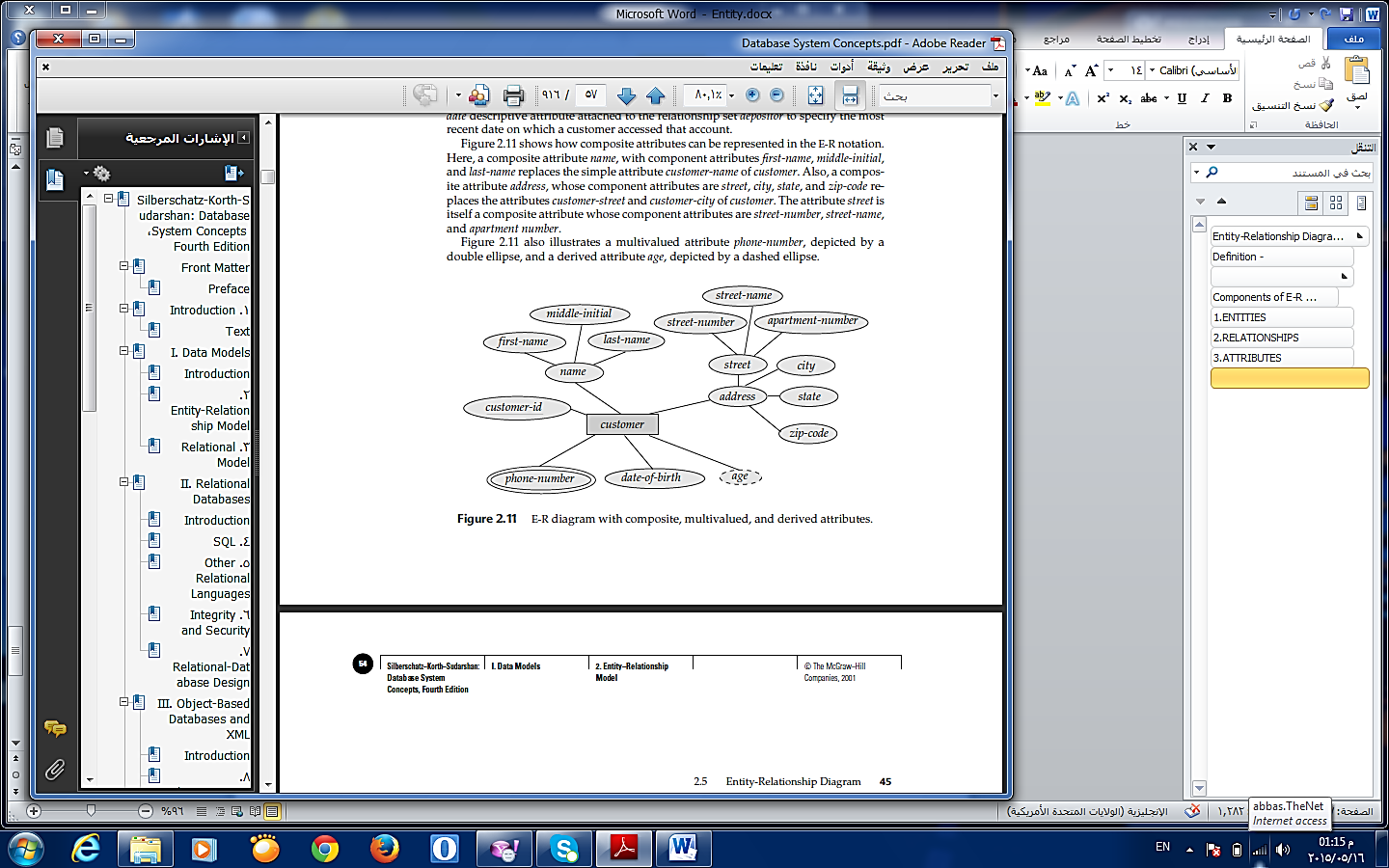
Returning to the E-R diagram of Figure 4.1, we see that the relationship set *borrower* is many-to-many. If the relationship set *borrower* were one-to-many, from *customer* to *loan*, then the line from *borrower* to *customer* would be directed, with an arrow pointingto the *customer* entity set (Figure 4.2a). Similarly, if the relationship set *borrower*

were many-to-one from *customer* to *loan*, then the line from *borrower* to *loan* would have an arrow pointing to the *loan* entity set (Figure 4.2b). Finally, if the relationship set *borrower* were one-to-one, then both lines from *borrower* would have arrows:



**Figure 4.2 Relationships. (a) one to many. (b) many to one. (c) one-to-one.**

Figure 4.3 also illustrates a multivalued attribute *phone-number*, depicted by a double ellipse, and a derived attribute *age*, depicted by a dashed ellipse.

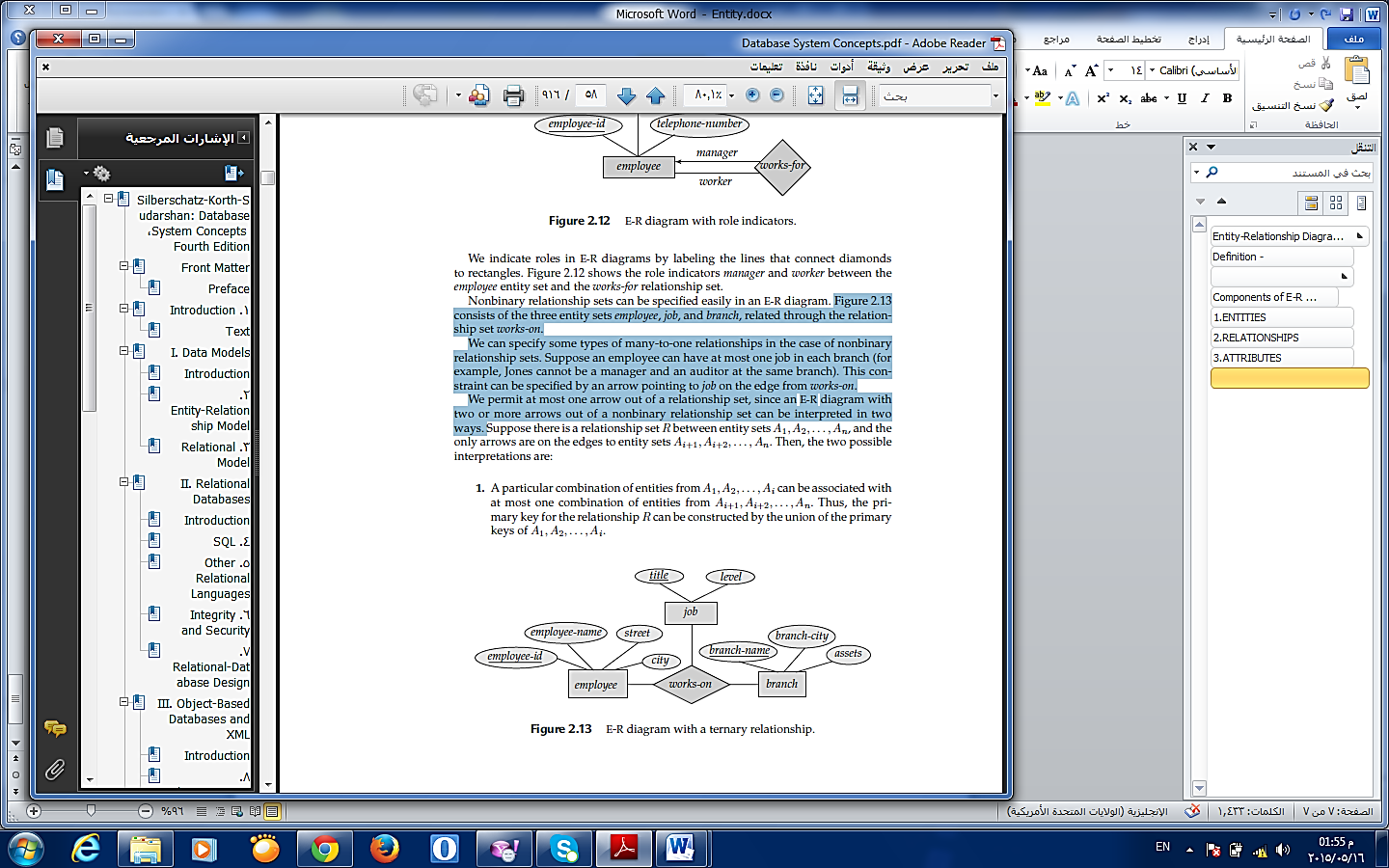


**Figure 4.3 E-R diagram with composite, multivalued, and derived attributes.**

Non-binary relationship sets can be specified easily in an E-R diagram. Figure 4.4 consists of the three entity sets *employee*, *job*, and *branch*, related through the relationship set *works-on*.

We can specify some types of many-to-one relationships in the case of non-binary relationship sets. Suppose an employee can have at most one job in each branch (for example, Jones cannot be a manager and an auditor at the same branch). This constraint can be specified by an arrow pointing to *job* on the edge from *works-on*.

We permit at most one arrow out of a relationship set, since an E-R diagram with two or more arrows out of a non-binary relationship set can be interpreted in two ways.

**Figure 4.4 E-R diagram with a ternary relationship.**

**4.5 Reduction of an E-R Schema to Tables**

We can represent a database that conforms to an E-R database schema by a collection of tables. For each entity set and for each relationship set in the database, there is a unique table to which we assign the name of the corresponding entity set or relationship set. Each table has multiple columns, each of which has a unique name.

Both the E-R model and the relational-database model are abstract, logical representations of real-world enterprises. Because the two models employ similar design principles, we can convert an E-R design into a relational design. Converting a database representation from an E-R diagram to a table format is the way we arrive at a relational-database design from an E-R diagram. Although important differences exist between a relation and a table, informally, a relation can be considered to be a table of values. The constraints specified in an E-R diagram, such as primary keys and cardinality constraints, are mapped to constraints on the tables generated from the E-R diagram.

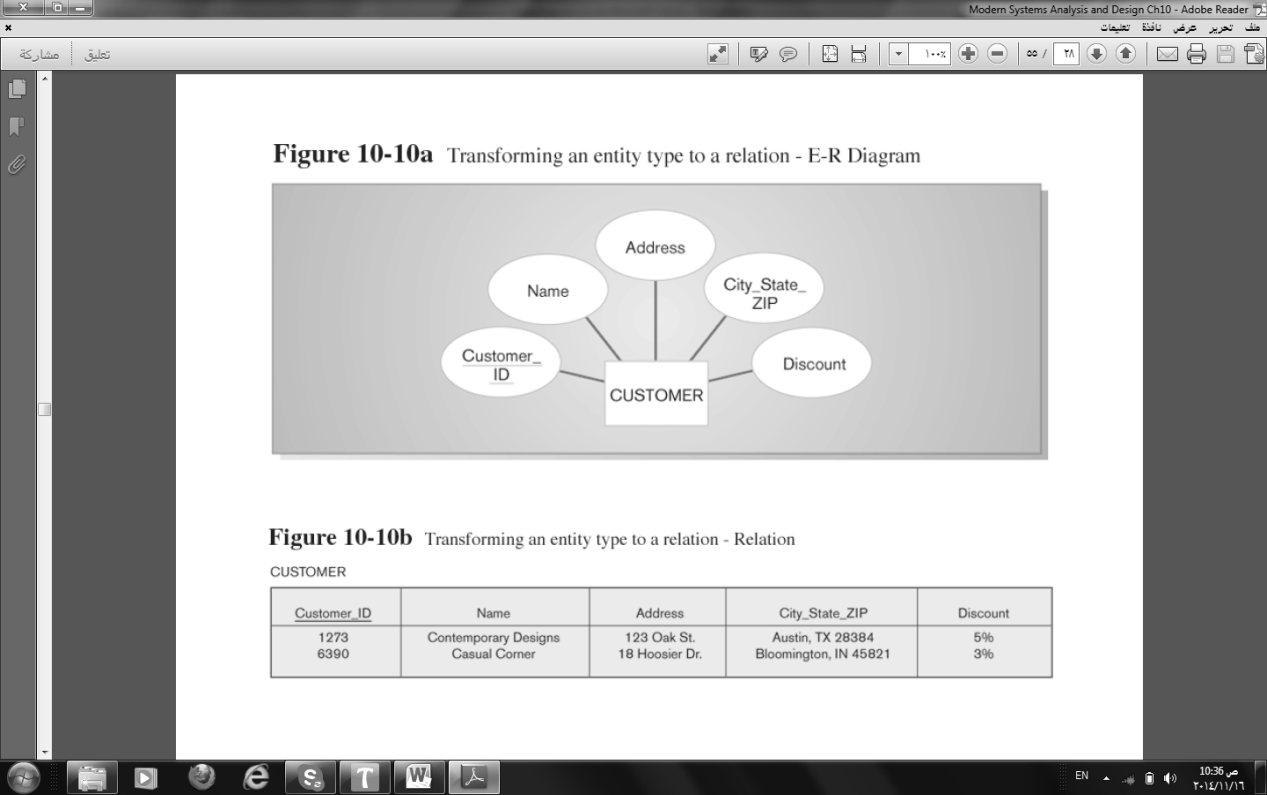
**Example :**

There is an entity:

**customer-schema=(customer-id,name,address,city-state-ZIP,discount).**

1.Transforming an entity to a relation – E/R Diagram.

2.Transforming an entity to a relation – relational .

****

**CUSTOMER**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Discount** | **City –State-Zip** | **Address** | **Name** | **Customer –ID** |
| 5% | Austin,TX2888 | 123 Oak St. | Contemporary Designs | 1273 |
| 3% | Bloomington ,IN5482 | 18 Hoosier Dr. | Casual Comer | 6390 |