**1. What do you mean by Entity- Relationship Diagram? Explain**

Entity-Relationship (ER) Model is based on the notion of real-world entities and relationships among them. While formulating real-world scenario into the database model, the ER Model creates entity set, relationship set, general attributes, and constraints. ER Model is best used for the conceptual design of a database. ER Model is based on:

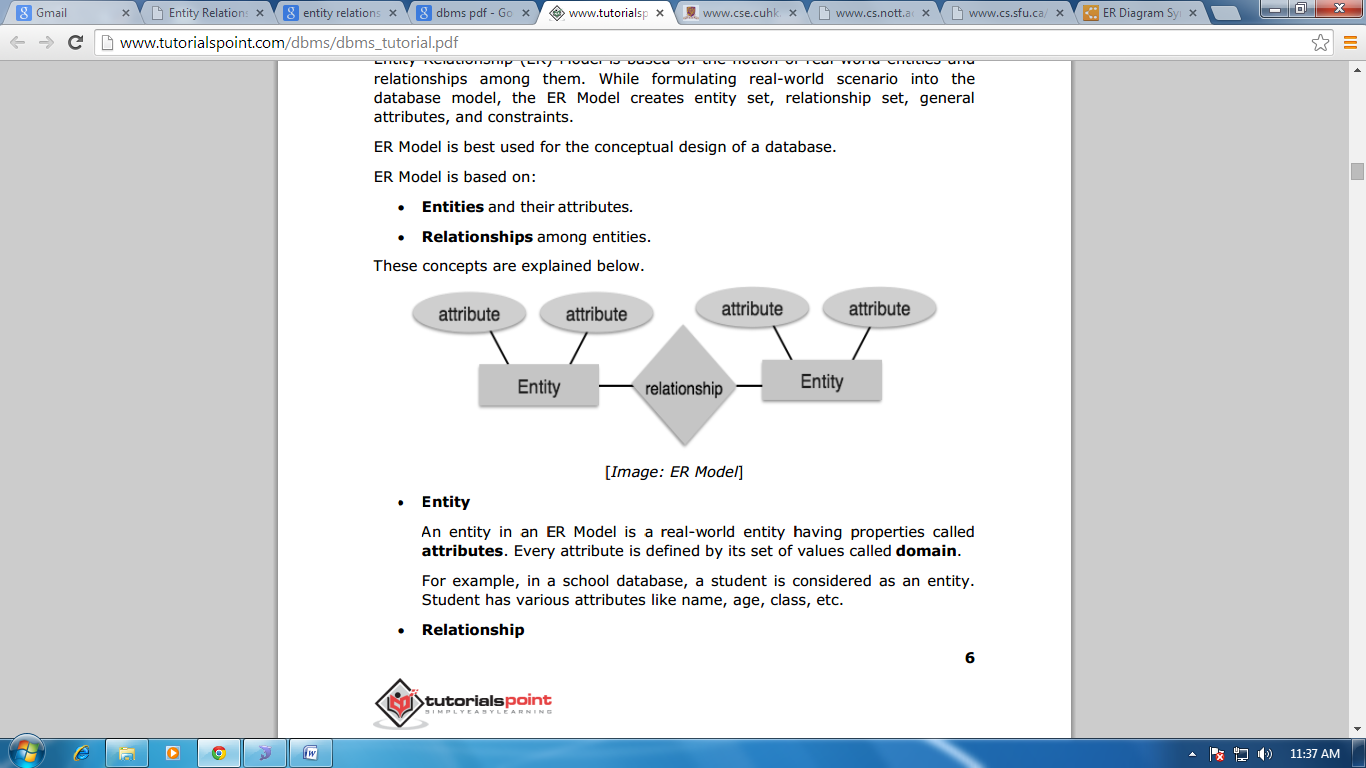
* Entities and their attributes.
* Relationships among entities.

Figure 1: E-R Diagram

**2. Define entity and give an example.** An entity in an ER Model is a real-world entity having properties called attributes. Every attribute is defined by its set of values called domain. For example, in a school database, a student is considered as an entity. Student has various attributes like name, age, class, etc.

**3. Explain the different between an entity class and an entity instance.** There is a difference between Entity Type and Entity Instance.

* Entity Type is a collection of entities that share common properties or characteristics
* Entity Instance is a single occurrence of an entity type

**4. Define attribute and its types.**

An attribute is a property or characteristic of an entity type that is of interest to an organization. Some attributes of common entity types include the following:

STUDENT = {Student ID, SSN, Name, Address, Phone, Email, DOB}

ORDER = {Order ID, Date of Order, Amount of Order}

ACCOUNT = {Account Number, Account Type, Date Opened, Balance}

CITY = {City Name, State, Population}

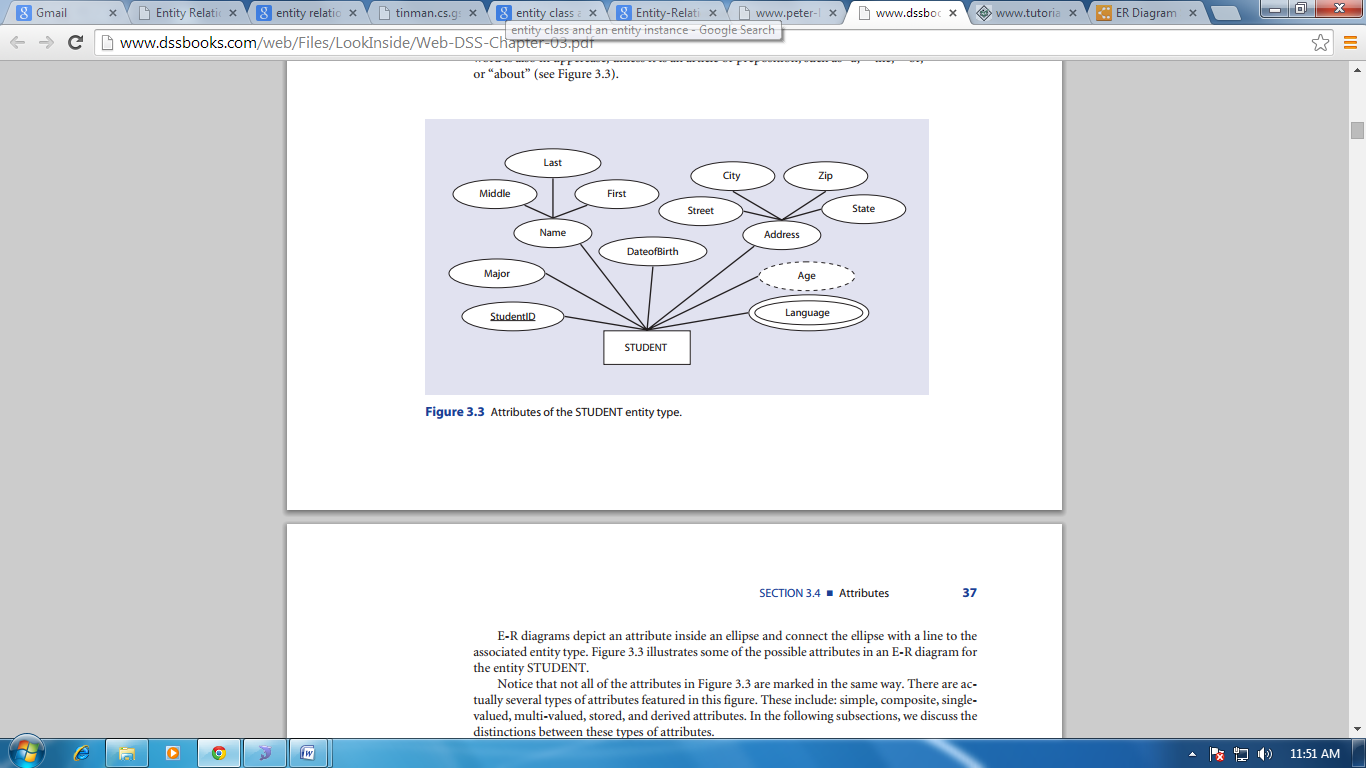


Figure : Attributes of the STUDENT entity type

* Simple and Composite Attributes

A simple or an atomic attribute, such as City or State, cannot be further divided into smaller

components. A composite attribute, however, can be divided into smaller subparts in which

each subpart represents an independent attribute. Name and Address are the only composite attributes. All other attributes, even those that are subcategories of Name and Address,

are simple attributes.

* Single-Valued and Multi-Valued Attributes

Most attributes have a single value for an entity instance; such attributes are called single-valued attributes. A multi-valued attribute, on the other hand, may have more than one value for an entity instance. Figure 2 features one multi-valued attribute, Languages, which stores the names of the languages that a student speaks. Since a student may speak several languages, it is a multi-valued attribute. All other attributes of the STUDENT entity type are single-valued attributes. For example, a student has only one date of birth and one student identification number. In the E-R diagram, we denote a multi-valued attribute with a double-lined ellipse. Note that in a multi-valued attribute, we always use a double-lined ellipse, regardless of the number of values.

* Stored and Derived Attributes

The value of a derived attribute can be determined by analyzing other attributes. For example, in Figure 2 Age is a derived attribute because its value can be derived from the current date and the attribute DateofBirth. An attribute whose value cannot be derived from the values of other attributes is called a stored attribute. As we will learn, a derived attribute Age is not stored in the database. Derived attributes are depicted in the E-R diagram with a dashed ellipse.

**5. What is derived attributes?**

An attribute whose value can be derived from the values of other attributes is called a derived attribute. The value of a derived attribute can be determined by analyzing other attributes. For example, in Figure 2 Age is a derived attribute because its value can be derived from the current date and the attribute DateofBirth. As we will learn, a derived attribute Age is not stored in the database. Derived attributes are depicted in the E-R diagram with a dashed ellipse.

**6. Define relationship and give an example.**

We define a relationship as an association among several entities. Consider, for example, an

association between customers of a bank. If customer Williams has a bank account number 523, then the quality of ownership constitutes a relationship instance that associates the CUSTOMER instance Williams with the ACCOUNT instance 523.

**7. Explain the difference between a relationship class and a relationship instance.**

**8. Define degree of relationship.** The number of entity sets that participate in a relationship is called the degree of relationship.

For example, the degree of the relationship featured in Figure 3 is two because CUSTOMER and ACCOUNT are two separate entity types that participate in the relationship. The three most common degrees of a relationship in a database are unary (degree 1), binary (degree 2), and ternary (degree 3).

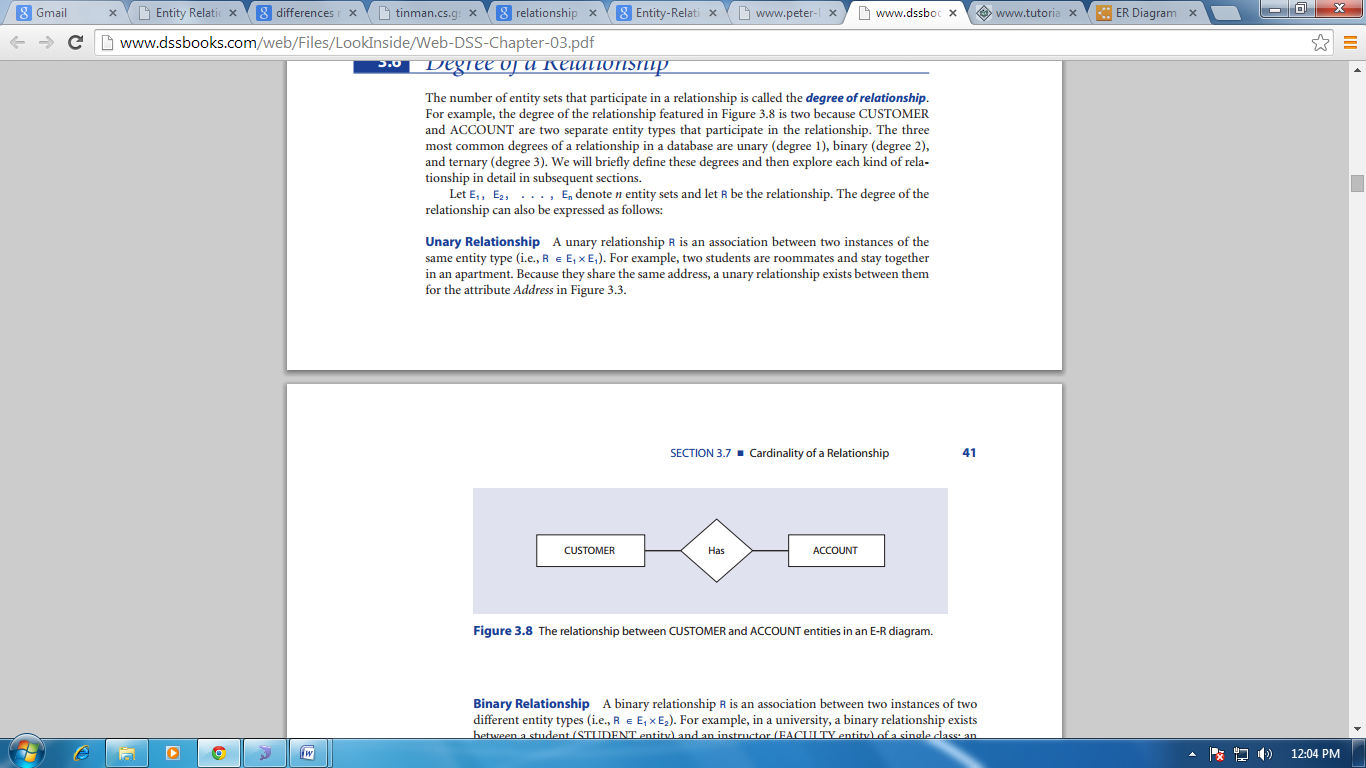


Figure The relationship between CUSTOMER and ACCOUNT entities in an E-R diagram

**9. List and give an example of the three types of binary relationships. Draw an E-R diagram for each.**

**Binary Relationship**: A binary relationship R is an association between two instances of two different entity types (i.e., R ∈ E1 × E2).

The three types of binary relationships include:

* Binary one-to-one relationships
* Binary one-to-many relationships
* Binary many-many relationships

Binary one-to-one relationships

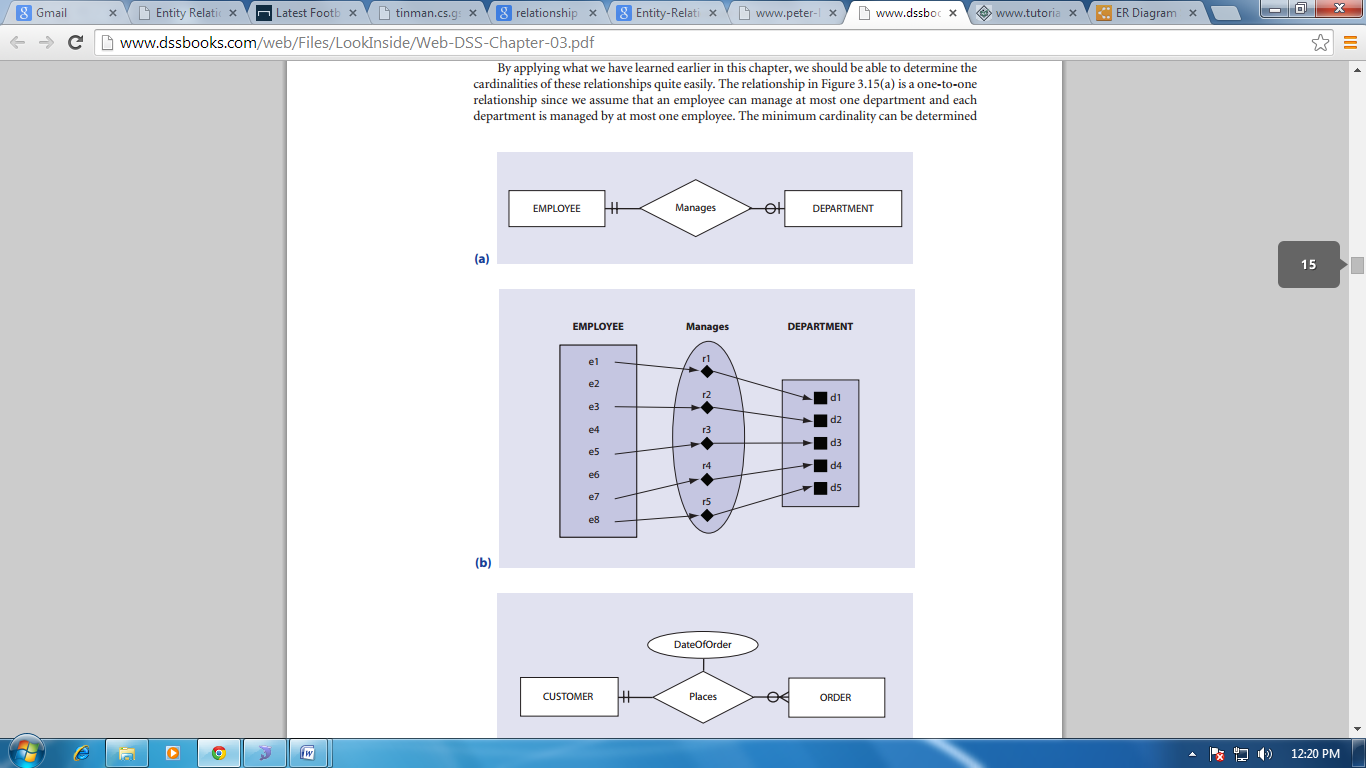
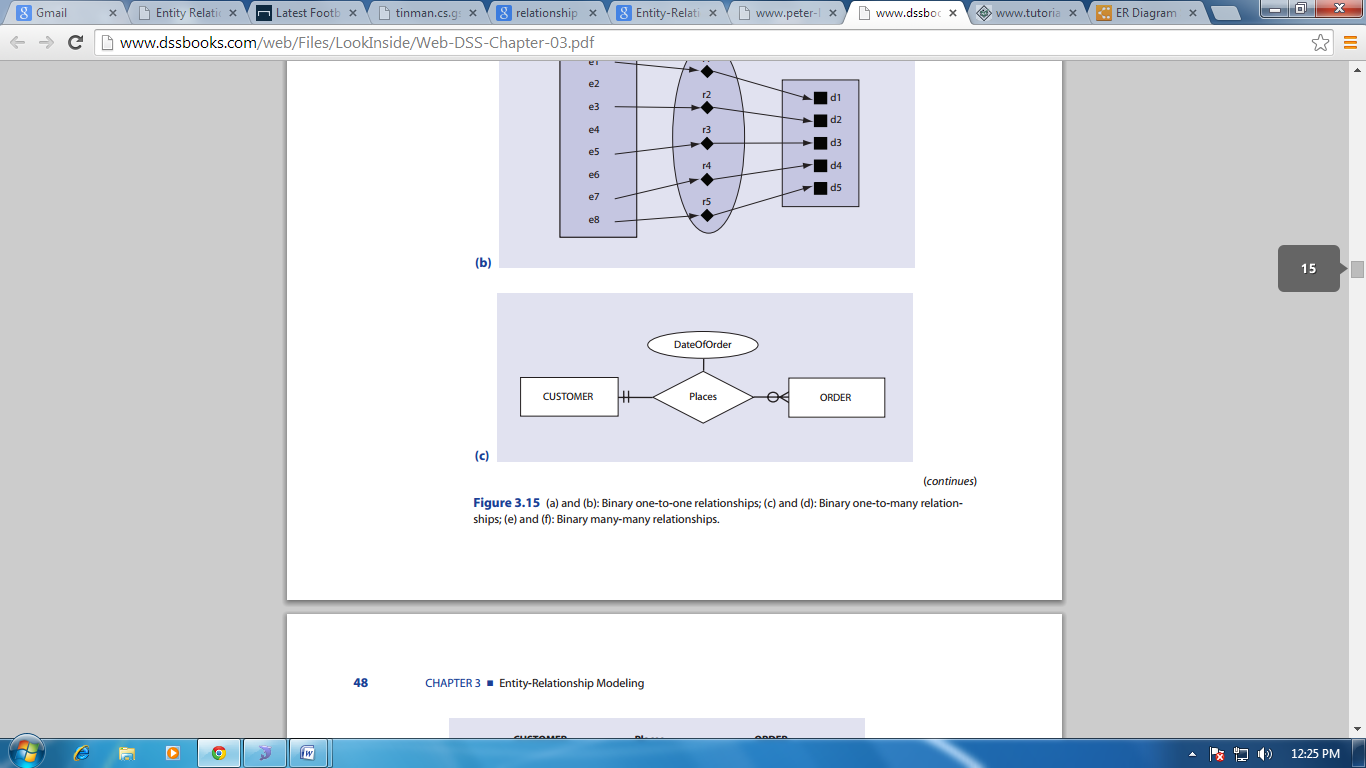
The relationship in Figure 4 is a one-to-one relationship since we assume that an employee can manage at most one department and each department is managed by at most one employee

Figure .1

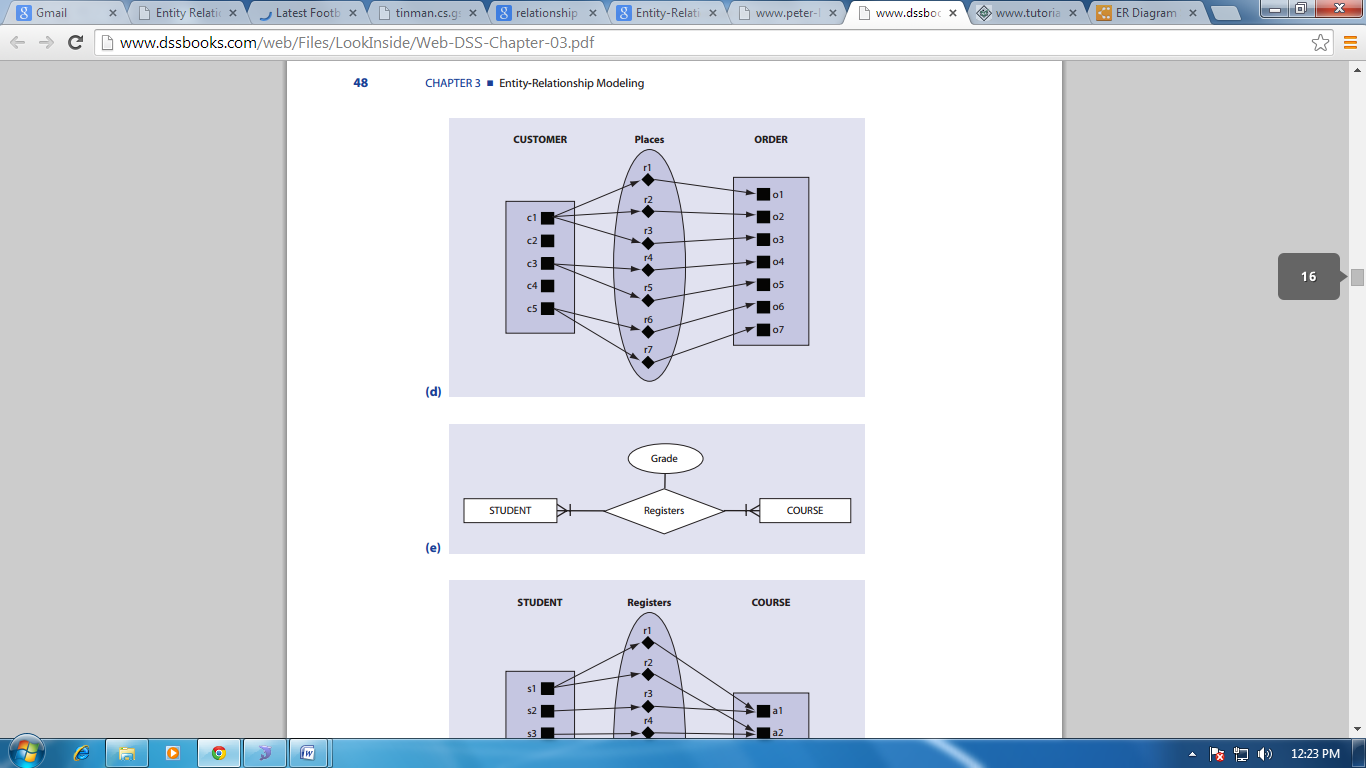
Binary one-to-many relationships

The binary one-to-many relationship represented in Figure 4.2 features a slightly different arrangement. While a customer can place several orders or may choose not to order at all, each order must be placed by exactly one customer.



Binary many-many relationships

Figure 4.3, illustrates a many-to-many relationship, featuring a minimum cardinality of zero and a maximum cardinality of many. In other words, each student can participate in many activities, a single activity, or no activity at all.



**10. Define the terms maximum cardinality and minimum cardinality.** Cardinality of a relationship means the ability to count the number of entities involved in that relationship. For example, if the entity types A and B are connected by a relationship, then the **maximum cardinality** represents the maximum number of instances of entity B that can be associated with any instance of entity A. The **minimum cardinality** between two entity types A and B is defined as the minimum number of instances of entity B that must be associated with each instance of entity A.

**11. Explain the distinctions among the terms primary key, candidate key and super key**