1. Define each of the following terms:

a. Entity type

The entity type is the fundamental building block for describing data structure with the Entity Data Model (EDM). In a conceptual model, an entity type represents the structure of top-level concepts, such as customers or orders. An entity type is a template for entity type instances, often corresponds to a table.

b. Entity-relationship model

An entity-relationship model (ER model) describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between entities (instances of those entity types).

c. Entity instance

Whereas an entity type represents an abstract category, an entity instance is a manifestation of an entity within that category. For example, Cell could be the entity type, but Cell1, Cell2, and Cell3 would represent the actual items within the network.

d. Attribute

Generally speaking, an attribute is a characteristic. In a database management system (DBMS), an attribute refers to a database component, for example, a table. It also may refer to a database field. Attributes describe the instances in the column of a database.

e. Relationship type

A relationship is a situation that exists between two relational database tables when one table has a foreign key that references the primary key of the other table. Relationships allow relational databases to split and store data in different tables while linking disparate data items. And relationship type is an association between entities.

f. Strong entity type

A strong entity type is an entity type whose existence doesn't depend on any other entity. Strong entity types have a key attribute. The key attribute helps in identifying each entity uniquely. It's represented by a rectangle. A relationship between two strong entities is represented by a diamond shape.

g. Weak entity

In a relational database, a weak entity is an entity that cannot be uniquely identified by its attributes alone; therefore, it must use a foreign key in conjunction with its attributes to create a primary key. The foreign key is typically a primary key of an entity it is related to.

h. Multivalued attribute

A multivalued attribute of an entity is an attribute that can have more than one value associated with the key of the entity. For example, a large company could have many divisions, some of them possibly in different cities.

i. Associative entity

An associative entity is a term used in relational and entity-relationship theory. A relational database requires implementing a base relation (or base table) to resolve many-to-many relationships. A base relation representing this kind of entity is called, informally, an associative table.

j. Cardinality constraint

Cardinality refers to the number of elements in a mathematical set. Thus, a cardinality constraint is a constraint that restricts the number of elements in a set.

k. Binary relationship

A Binary Relationship is a relationship between two different Entities, i.e., it is a relationship of the role group of one entity with the role group of another entity.

There are three types of cardinalities for Binary Relationships:

1. One-to-One
2. One-to-many
3. Many-to-Many

l. Unary relationship

A unary relationship (also called recursive) is a relationship that exists between occurrences of the same entity sets. In this kind of relationship, the primary and foreign keys are the same, but they represent two entities with two different roles.

m. Business rule

A business rule is a statement that imposes some form of constraint on a specific aspect of the database, such as the elements within a field specification for a particular field or the characteristics of a given relationship.

2 Why must some identifiers be composite rather than simple?

An identifier is an attribute (or combination of attributes) whose value distinguishes individual instances of an entity type. Usually, a simple attribute isn't unique for all occurrences of an entity type (for example, FlightNumber for airline flight). Rather, a combination of several simple attributes is needed to uniquely identify the entity instance (for example, FlightID and FlightDate would make the instance unique)