ER Model - Basic Concepts

The ER model defines the conceptual view of a database. 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.

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 school; likewise a Teachers set may contain all 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.

Types of Attributes

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.

Composite attribute – Composite attributes are made of more than one simple attribute. For example, a student's complete name may have first_name and last_name.

Derived attribute – Derived attributes are the attributes that do not exist in the physical database, but their values are derived from other attributes present in the database. For example, average_salary in a department should not be saved directly in the database, instead it can be derived. For another example, age can be derived from data_of_birth.

Single-value attribute – Single-value attributes contain single value. For example – Social_Security_Number.

Multi-value attribute − Multi-value attributes may contain more than one values. For example, a person can have more than one phone number, email_address, etc.

These attribute types can come together in a way like −

simple single-valued attributes

simple multi-valued attributes

composite single-valued attributes

composite multi-valued attributes

Entity-Set and Keys

Key is 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.

Super Key − A set of attributes (one or more) that collectively identifies an entity in an entity set.

Candidate Key − A minimal super key is called a candidate key. An entity set may have more than one candidate key.

Primary Key − A primary key is one of the candidate keys chosen by the database designer to uniquely identify the entity set.

Relationship

The association among entities is called a relationship. For example, an employee works_at a department, a student enrolls in a course. Here, Works_at and Enrolls are called relationships.

Relationship Set

A set of relationships of similar type is called a relationship set. Like entities, a relationship too can have attributes. These attributes are called descriptive attributes.

Degree of Relationship

The number of participating entities in a relationship defines the degree of the relationship.

Binary = degree 2

Ternary = degree 3

n-ary = degree

Mapping Cardinalities

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.

One-to-one – One entity from entity set A can be associated with at most one entity of entity set B and vice versa.

One-to-many – One entity from entity set A can be associated with more than one entities of entity set B however an entity from entity set B, can be associated with at most one entity.

Many-to-one – More than one entities from entity set A can be associated with at most one entity of entity set B, however an entity from entity set B can be associated with more than one entity from entity set A.

Many-to-many – One entity from A can be associated with more than one entity from B and vice versa.

How to create an ER Diagram

While ERDs may seem hard, they’re not as tricky as they appear. Entity-relationship diagrams can easily be built with the shapes included in a free Lucidchart account. Follow these steps to begin diagramming.

Make an ERD

Identify the components

Entity-relationship diagrams are incredibly useful, and you can easily create one of your own by following these simple steps.

1. Determine the entities: Entities are typically nouns such as car, bank, student, or product.

In an ER Diagram, entities are the most important parts. To proceed, we will be creating a conceptual ER diagram of a simple system in which a student registers for a course that is taught by a professor. Check out this awesome tutorials to review ER Diagram Shapes. In this example, the three entities are “Student,” “Course,” and “Professor.”

2. Identify the relationships: Relationships highlight how entities interact with each other.

Relationships are typically verbs such as “buys,” “contains,” or “does.” In our example, the relationships “Registers for” and “Teaches” effectively explain the interactions between the three entities.

3. Add attributes: Attributes show specific characteristics of an entity, refining what information is important to the model.

In an ER Diagram, attributes are necessary to model what characteristics will be included with each entity. Attributes such as “IDNumber,” “Name,” and “SKU” are common attributes.ER Diagram Attribute Shapes

4. Complete the diagram

Organizing the ERD in a logical way is incredibly important to increase comprehension. The main purpose of entity-relationship diagrams is to model a complex database, so learning how to create simple, logical ERDs is key.