DBMS LAB 06 MATERIAL

Prepared by:
Mohammad Anas Jawad
Lecturer, IUT CSE

Department of Computer Science and Engineering
Islamic University of Technology
March 10, 2020

Contents

1 THE ER MODEL

ER Model or the Entity Relationship Model is the most popular modelling scheme in database design. The architecture of a database is initially planned and designed by using ER diagrams, and then the final design is implemented.
Similar to software design, a database designer has to talk to appropriate individuals to get the list of requirements needed for a project. When describing these projects, it is easier to explain things by using diagrams instead of using technical terms like tables, attributes and so on as most people are not familiar with the terms used in computer science. Once all the requirements have been clearly conceptualised using the ER Model, a database designer can proceed to the actual implementation of the project.
Let's look at the things that make up an ER model and we can get a clearer picture on the whole concept.

1.1 Entity

An entity is a "thing" or "object" in the real world that is distinguishable from all other objects. For example, each person in a university is an entity.

1.1.1 Entity Set VS Entity

When we talk about entities, we first have to understand entity sets. Entity set is a generic representation or description of an entity. In other words, entity set or entity type is the heading or schema, while entity is an instance of that entity set. Entities are also called ***extensions of entity sets***. Let's look at an example:
Entity set: *Student(ID, Name, Age, Semester)*
Entity: (10101, 'Sam', 24, 'Final')
Note: Whenever people talk about their requirements in a project, they will use different words that might be nouns, verbs etc. A good way to initially get a grasp on the entities of a project are by distinguishing its nouns. The nouns usually correspond to different entities that will be present in the final database design.

1.2 Attributes

Attributes are nouns that describe the entities. For example, in the Student entity, the attributes are ID, Name, Age and Semester. Attributes can be of different types depending on the design of the database.

1.2.1 Composite VS Simple-valued attributes

Composite attributes are those that can be split into multiple parts. For example, the Name attribute caa be split into First Name, Middle Name and Last Name.
Simple-valued attributes cannot be split further to produce more attributes. For example, the attribute 'First Name' cannot be split further.
From this we understand that a composite attribute is essentially a combination of simple attributes.

1.2.2 Single-valued VS Multi-valued Attributes

Some attributes can have multiple values. For example, the attribute 'Phone Number' may have multiple values for an individual user. However, the attribute 'Age' can store only a single value for a particular user. So, Age is a single-valued attribute.

1.2.3 Stored VS Derived attributes

Sometimes, some attributes may need to be derived from other attributes. Consider the following scenario: You have an attribute called 'DOB' that stores the date of birth of a person. Now, for this DOB, you can calculate that person's Age. Then, Age essentially becomes a derived attribute. On the other hand, stored attributes are basically all other attributes which we store directly in the database. So, DOB is a stored attribute.

1.2.4 Complex Attributes

An attribute can be a combination of different types. For example, the attribute 'Address' can be a composite and multi-valued attribute simultaneously. These attributes are called complex attributes.

1.3 Relationship

A relationship is an association among entities. As a database designer, after completing your requirement analysis phase, you will have a description of the requirements at your disposal. The verbs in the description will usually give an overview of the relationship or association that exists among the entities of your project.
Similar to Entity Sets, Relationship Sets are generic representation or description of Relationships. That is, each Relationship is an instance of a Relationship Set.
The relationships are usually given meaningful names based on the scenario.

2 EXAMPLE SCENARIO

Every employee works under a certain department. Each employee has an ID, name and salary. A department may have multiple employees. A newly formed department may have zero employees. Each department must have a name and must be located at some particular building within the campus.

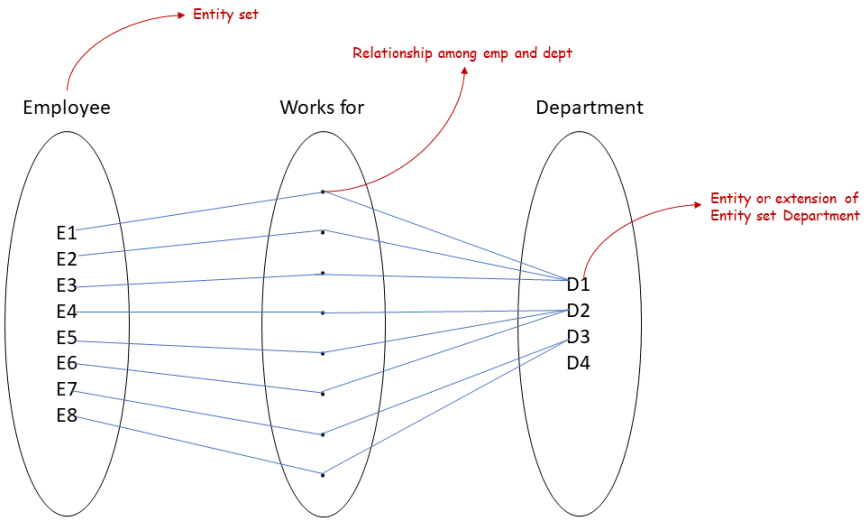


Figure 1: Depiction using set theory

3 PROPERTIES OF RELATIONSHIP

3.1 Degree

Degree of a relationship denotes the number of entities participating in a relationship. In our scenario, two entities, Employee and Department participate in the relationship. Hence, its degree is 2 i.e. it's a binary relationship.

3.2 Cardinality Ratio constraint

Cardinality ratio denotes the maximum number of relationships an entity can participate in. Let's look at our example scenario to get a clearer picture.
Each of the employees must be enrolled under a single department, i.e. an Employee can work under a maximum of 1 Department. So, the cardinality of Employee is 1. On the other hand, a department can have multiple Employees working under it. So, the cardinality of Department can be denoted as M(more than 1).

3.3 Participation or Existence constraint

Participation denotes the minimum number of relationships an entity can participate in. It is also sometimes referred to as minimum cardinality.
In our example scenario, each Employee must work under a Department. So, the participation of Employee is 1. However, a Department may or may not have any Employee. So, the participation of Department is 0. Hence, the participation of Employee in the relationship set WorksFor is a **total relationship**, whereas the participation of Department in the relationship set WorksFor is a **partial relationship**.
Note: All these properties can be understood by looking at the problem description.

4 ER REPRESENTATION

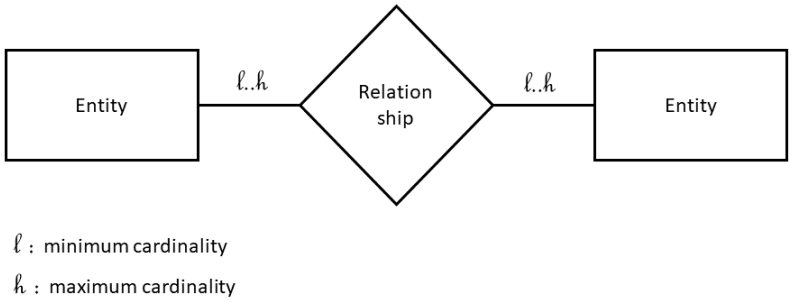


Figure 2: Notations of ER Modelling

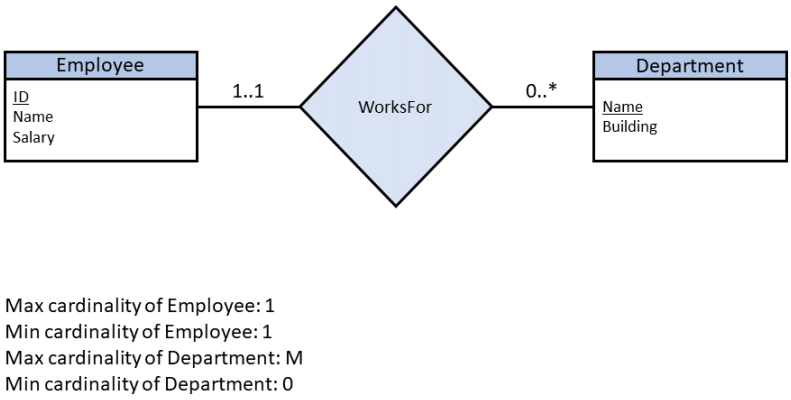
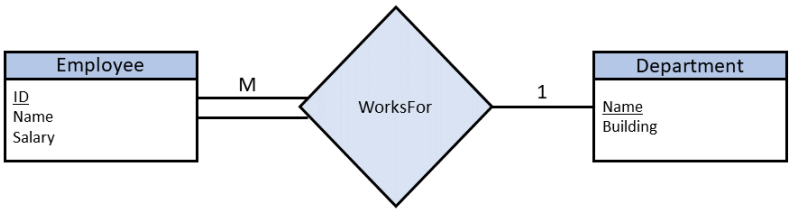


Figure 3: Minimum-Maximum representation of example scenario



Max cardinality of Employee: 1
Min cardinality of Employee: 1
Max cardinality of Department: M
Min cardinality of Department: 0

Figure 4: Single line-Double line representation of example scenario

5 REPRESENTATION OF DIFFERENT TYPES OF ATTRIBUTES

Simple valued attributes will be represented using a single oval.

Composite attributes (e.g. address) will be represented by a single oval, surrounded by other ovals representing its component attributes.

Multivalued attributes (e.g. phone number) will be represented using double oval.

Derived attributes (e.g. age) will be represented by dotted ovals.

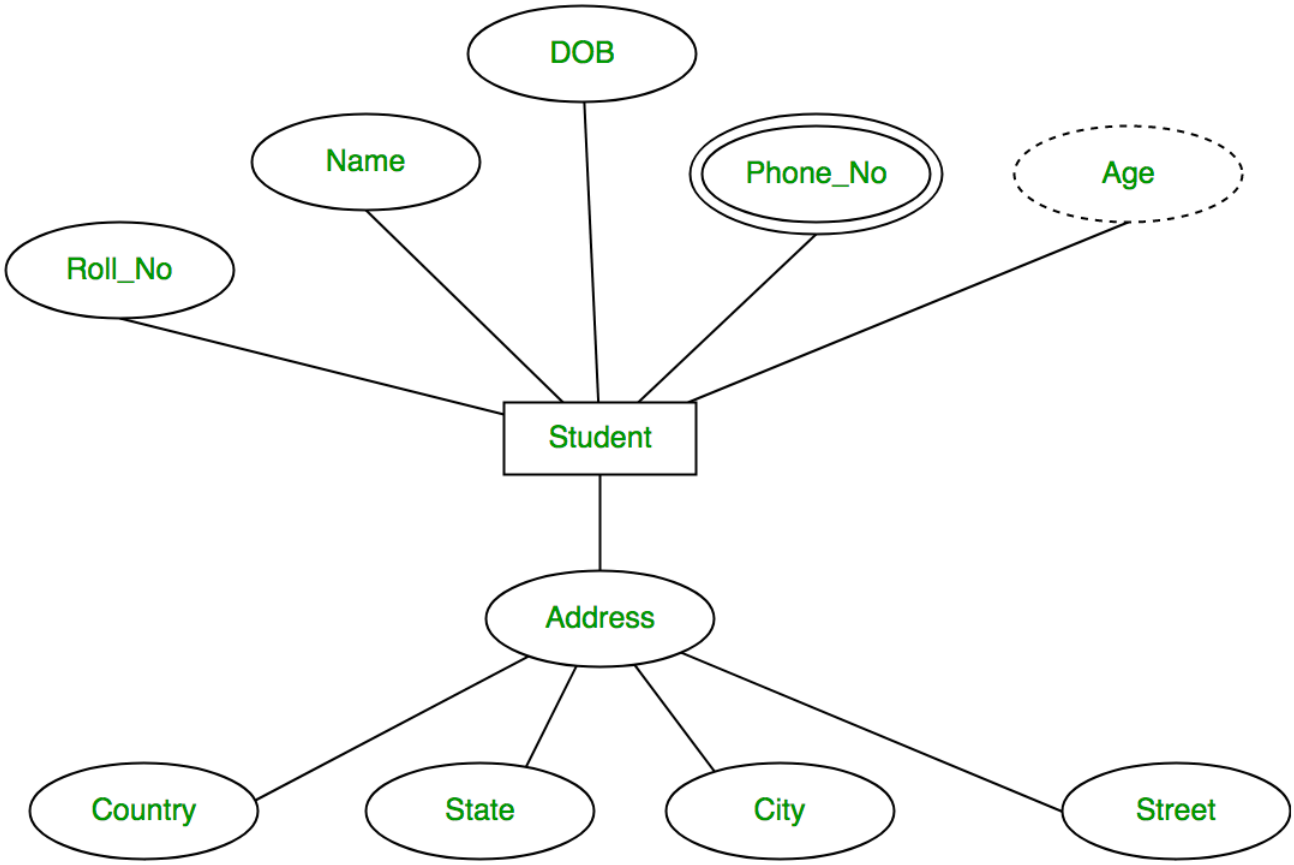


Figure 5: Representation of multiple types of attributes [Source: geeks4geeks]