Entity Relationship Model

Objectives:

At the end of the laboratory session, the student is expected to:

- Identify the different components of an ER Diagram
- Identify the different types of attributes
- Identify the cardinality of an entity being modeled
- Differentiate partial from total participation
- Create an ERD for a given problem

In programming intensive courses such as CMSC 11 and CMSC 21, our instructors would usually tell us that we should first create a pseudo code or a flowchart before coding the solution. The reason for this is simple: to separate the solving of the problem from implementation of the solution for the problem. This manner simplifies the means of solving the problem as students have the tendency to intertwine the solution and the implementation.

The same is true for CMSC 127. One of the usual expectations from you after taking CMSC 127 is for you to be able to demonstrate mastery in performing SQL queries. But before that, you need to be able to implement a schema (do not bother about this term for the moment) using some modeling tools that you will learn from the course. One of those is the ER Model.

ER stands for Entity-Relationship. The concept of ER Model is very simple to understand. In a gist, the **ER Model** explains to us that in a given system that needs to be modeled, there are entities that need to be identified. These entities interact with one another (either interacting with some or all of the entities), thus there exist a relationship with one or among the entities. The ER Model is visualized graphically using an Entity Relationship Diagram (ERD).

So what is an entity then? An **entity** is an object or item that exists on its own and has attributes. An entity is usually a noun. Take note that an entity is not limited to tangible objects as you would see later. In a strict sense, an entity is an instance of what is called as the entity type. An **entity type** is said to be a set of entities having the same set of attributes.

Suppose that you are asked to identify entity types in a bank. Some of the possible answers include depositors, bank managers and clerks. An example of a non-tangible entity type would be a bank account.

One additional note that you should remember is that entities (and entity types) in the ER Model are always expressed in singular form. They are also written using capital letters. Thus, the above example should then be DEPOSITOR, BANK MANAGER, CLERK and BANK ACCOUNT.

To better understand the difference of the two, let us go back to the example presented. We can say that you, I, and an ICS faculty member are entities under the entity type DEPOSITOR.

Self Assessment Question(s)

Identify at least five (5) entity types in

1. a university
2. a clothing factory
3. facebook

We have said so much about entities and entity types but have not yet mentioned about the attributes. Attributes are characteristics of an entity that defines its state. It also shows how an entity differs or bears similarity with another entity.

Using the bank example, we can say that a DEPOSITOR has a Name, a Gender, a Birthday, an Age, a Work address, a couple of Contact numbers and a couple of Home Addresses. Notice that the while the entities/entity types are written in capital letters, the attributes are written with the first character in capital. Similar to entities, attributes must be also expressed in their singular form (Contact number and Home Address).

There are different types of attributes according to how you classify them. Here are some of the common classifications of attributes.

First is what we call **atomic attribute**. This attribute cannot be broken down into smaller attributes and is simple on its own. It can also assume one value at an instance. Referring to our previous example DEPOSITOR, Gender is an example of an atomic attribute. Age is also an example of an atomic attribute but is special on its own way as we will see later. The other type of attribute is the **composite attribute**. This attribute is composed of atomic attributes. An example of this would be Work address. As you know, an address is composed possibly of a building name, building number, baranggay, city or town, province and others. A special attribute called **derived attribute** happens when the value of an attribute is dependent, based or computed using another attribute. An example based on the DEPOSITOR is the Age. The Age can be computed using the Birthday attribute. It is important to note and to stress that if the Birthday attribute is not considered or removed, then Age will no longer be a derived attribute.

Self Assessment Question(s)

Assuming STUDENT, COURSE and FACULTY as your entities (more correctly, entity types), name one atomic, derived and composite attribute.

There is also what is called as the **single-valued** and **multi-valued attributes**. **Single-valued attributes** refer to attributes that can hold only at most one value in an instance while **multi-valued attributes** can hold more than one values in an instance. Age is a single-valued attribute and Contact number is a multi-valued attribute. A DEPOSITOR's Hair color can also be considered as a multi-valued attribute.

We also have what we call **complex attributes** which are attributes that are both composite and multi-valued at the same time. For example, if you happen to have multiple homes, then you

have multiple addresses. In this example, Home Address can be a complex attribute otherwise, it will just be a composite attribute.

Self Assessment Question(s)

Identify which of the following attributes of a STUDENT is multi-valued. Justify why they are multi-valued.

1. Girlfriend/Boyfriend
2. Student name
3. Student nickname
4. High school attended

Entities also have classifications but unlike attributes, entities can either be a **strong entity** or a **weak entity**. A **strong entity** has a defining attribute (termed as a key attribute) while a **weak entity** has no defining attribute.

A **defining attribute** is an attribute that can uniquely identify an entity. A defining attribute is also called **key attribute**. For example, a STUDENT entity type can have a Student number as its defining attribute. This is because we know that each STUDENT has his/her own Student number and that we know that they are unique.

Usually, an entity should be strong but there are cases where an entity is weak. If the entity is weak, instead of a key attribute, a **partial key** is selected. A partial key is selected from the set of attributes which is the most possible attribute to identify an entity.

So far, we have talked about entities and attributes. We have not yet discussed that an entity interacts to another entity and that they have a **relationship**. Relationships are expressed as verbs and are written in capital letters. For example, a DEPOSITOR OWNS a BANK ACCOUNT, STUDENT ENROLLS in a COURSE and a FACULTY (member) TEACHES a COURSE.

ER Constraints

There are two types of constraints. They are the **cardinality** and the **participation constraints**.

Cardinality constraint refers to the number of times that an entity can participate in a relationship with another entity. For example, we can see that in a usual university setting, a STUDENT can HAVE many SUBJECTs, a STUDENT can be admitted to multiple DEGREE PROGRAMs, and a STUDENT HAS only one (thesis) ADVISER. There are four basic cardinalities: **one to one** (1:1), **one to many** (1:N), **many to one** (N:1) and many **to many** (M:N). If the cardinality is many to many and we are sure that the two entities have the same number, it is expressed as N:N.

There are instances however, that the basic cardinalities do not suffice to describe a real life relationship among entities. For example, suppose we say that an ADVISER can ADVISE many

ADVISEEs, we can set a cardinality of one (1) to ADVISER and many (N) to the ADVISEE. However, suppose that we became more specific such that we specified that an ADVISER can have at least one ADVISEE, how can we do that using the basic cardinalities? Obviously, we cannot. In such cases, the **Min-Max Notation** is used.

In Min-Max Notation, the modeler is given freedom to provide a minimum value and a maximum value for the number of times the entity participates in the relationship following this format: (min, max). In our given example about the ADVISER and the ADVISEE, we can have something like this 1:(1,N).

The **participation constraint** on the other describes whether or not an entity is required to participate in a relationship with another entity. This constraint is sometimes called the **minimum cardinality constraint** because it describes the minimum cardinality for an entity E for a relationship R. There are two types of participation: **total** and **partial**. A **total participation** happens when an entity E1 is required to participate in a relationship R with an entity E2 while **partial participation** happens when an entity E1 is not required to participate in a relationship R with entity E2.

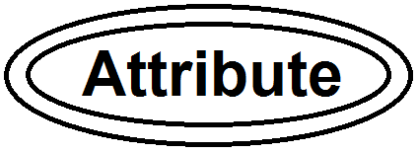
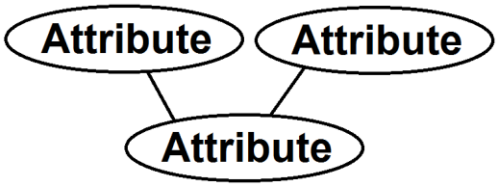
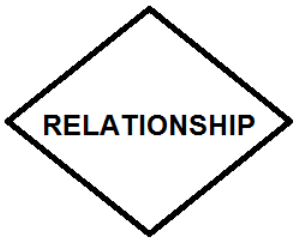
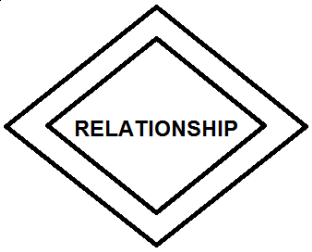
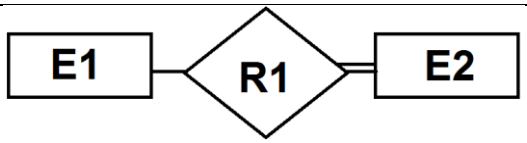
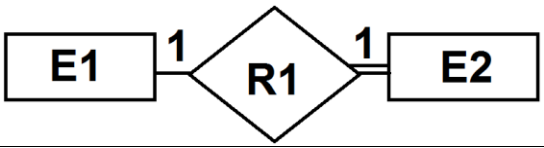
For example, if we describe a situation that an ADVISER ADVISEs an ADVISEE and that it is possible that an ADVISER can ADVISE zero (0) ADVISEE yet the ADVISEE must be ADVISEd by an ADVISER, we can say that in the relationship ADVISE, the ADVISER is of partial participation and the ADVISEE is of total participation. If you are having problem with this one, do not worry. Participation constraint is quite tricky and requires experience. However, for lecture/laboratory exercises and activities, we will be providing the details to you.

The ER Diagram

Here is the interesting part in the ER Model: you will have to draw your model. The ER Model is visualized through the **Entity Relationship Diagram (ERD)** or a Class Diagram. We will first focus on ERD since there is a separate chapter for Class Diagram.

An ERD uses a set of symbols that are associated to the concepts that have been previously presented. Below is the summary of the symbols used:

Symbol	Meaning/Notes
	Entity types are represented by a rectangle enclosing the entity type. Entity types must be in uppercase.
	A single-valued atomic attribute is enclosed in an oval and attached to the entity. The first letter of the attribute is capitalized.
	A derived attribute is enclosed in a dashed oval and attached to the entity. The first letter of the attribute is capitalized.

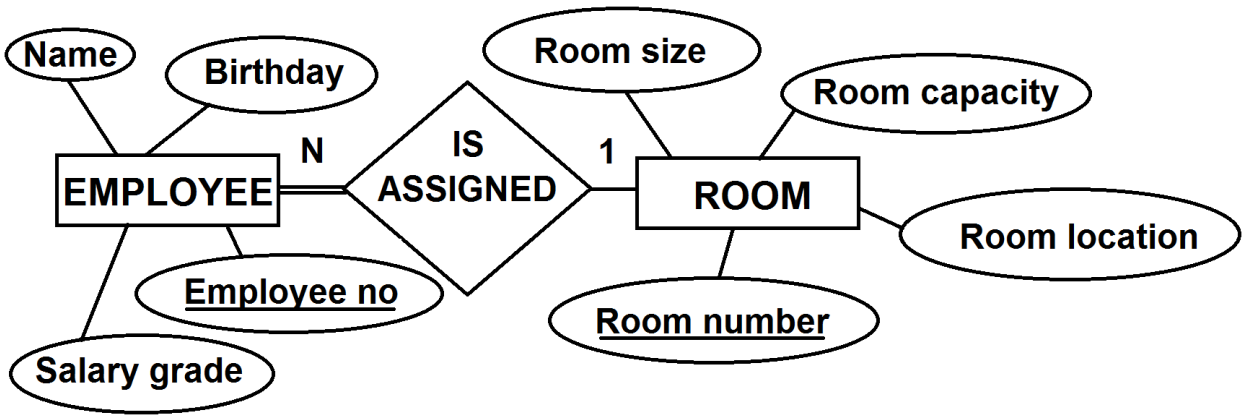
	A multi-valued attribute is enclosed in a double oval and attached to the entity. The first letter of the attribute is capitalized.
	A composite attribute contains the attribute with a stem of atomic attributes. The first letters of the attributes are capitalized.
	Key Attribute
	Partial Key
	A relationship is represented by a diamond. Relationships are verbs and are written in capital letters.
	An indentifying relationship exists when one of the entities in a relationship is a weak entity.
	A weak entity is represented by a double box.
	A total participation is represented by two lines connecting the entity and the relationship A partial participation is represented by one line connecting the entity and the relationship
	The cardinality is written on top of the line connecting the entity and the relationship

Here is the interesting part in the ER Model: you will have to draw your model. The ER Model is visualized through the **Entity Relationship Diagram (ERD)** or a Class Diagram. We will first focus on ERD since there is a separate chapter for Class Diagram.

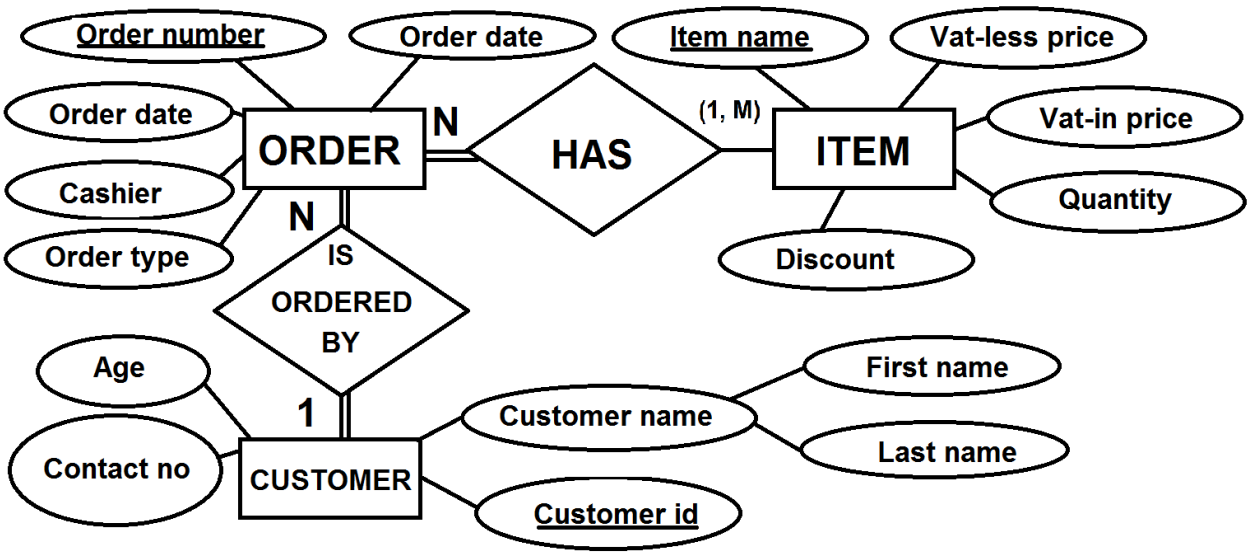
An important note that must be established at this point is that we assume that all the relationships described in this manual, unless otherwise specified, are **binary relationships**. This means that in every relationship, there should only be at two entities involved.

In order to provide you a better understanding of this chapter's concept, we will proceed to providing you a couple of examples. Examine how we arrived at the answers/solutions provided.

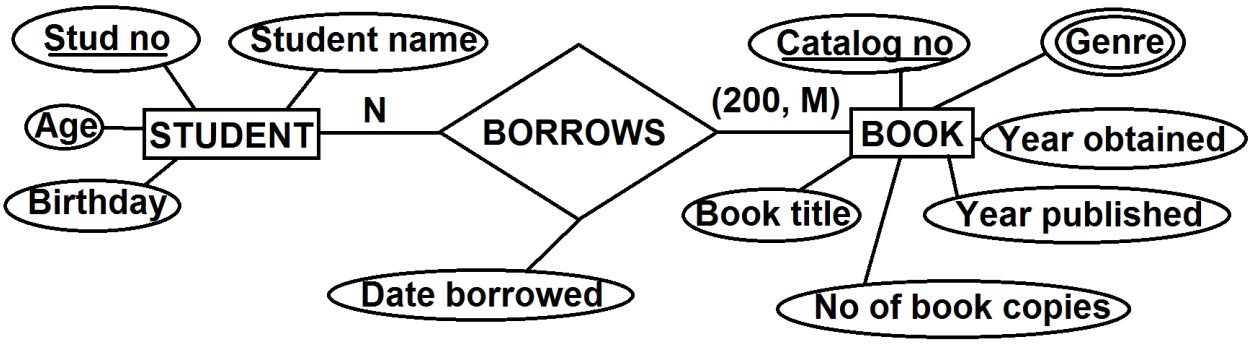
EXAMPLE 1: An employee is assigned to a room in the company. An employee has a name, birthday, salary grade and an employee number which is unique. A room has a room number, room size, room capacity, and room location. A room number can identify the room. A number of employees can be assigned to a room. An employee must be assigned to a room but a room is not required to be assigned with an employee. An employee can be assigned to only one room.



EXAMPLE 2: A fast-food restaurant keeps track of the orders made by a customer. An order is described by its unique order number, order date, order time, cashier who punched the order into the POS terminal and the type of order (dine in, take out, special or bulk). An order has items from their menu. An item is described by its item name, vat-less price, vat-in price, quantity and whether the item is on some form of discount or promo. The item name is unique. A customer has a name composed of first name and last name, age, and contact number and a customer id. Customer id is unique. An order needs at least one order. An item can be included in any order. A customer always has an order and an order is always made by a customer. A customer can make multiple orders.

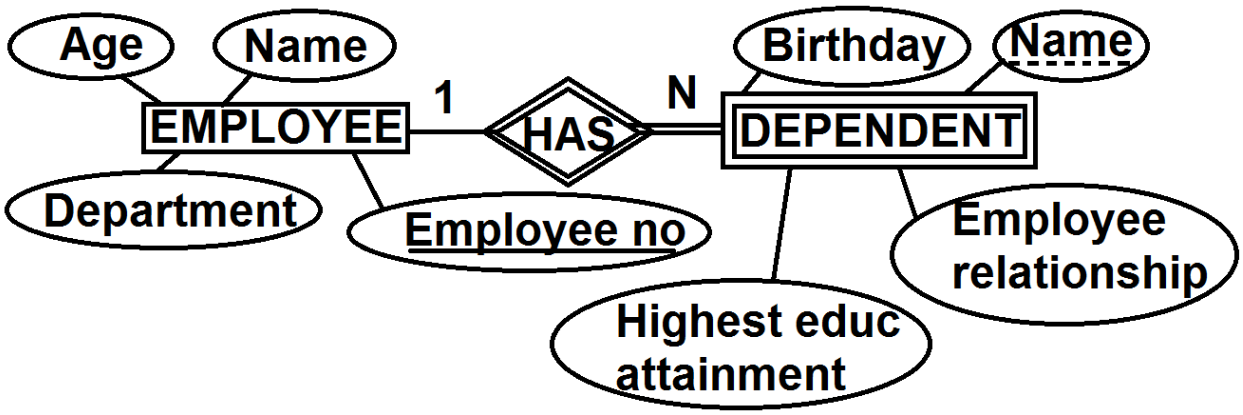


EXAMPLE 3: The UPLB Library keeps track of all the students who borrow books from the library. A student can borrow only up to 200 books and is possible to borrow no books at all. It is possible that a book is not borrowed. A book can be borrowed by more than one employee. A book has a book title, a catalog no which is unique, year published, year when the library obtained a copy of the book, number of book copies (assume that for a single book title, they are only obtained once e.g., Happy Rotter book was obtained in 2003 with 10 copies. The case Happy Rotter with 10 copies obtained in 2003, 2005 and other additional years is not possible) and genre. It is possible for a book to belong to many genres. A student has a unique 9 digit student number, a student name, birthday and age. We also keep track the date when a book is borrowed by a student.



EXAMPLE 4: An employee is entitled with some form of benefits. These benefits often extend to dependents. A dependent of an employee is defined as a either a parent with at age of at least 64 years old or a child of at most 18 years old. A dependent is described by his/her name, birthday, relationship to the employee and highest educational attainment. The employee is described by his/her name, employee no, age and department. The employee no is unique. It is possible for an employee to have no

dependents. A dependent must always have an employee to depend on. A dependent depends only to one employee.



EXERCISES

Draw an ERD for the following problems.

1. A chef cooks different dishes served to his or her guest in the restaurant. A dish is described by its name, dish origin, main ingredient, number of calories and its dish type (entree, main dish, appetizer, etc) and price. One or more dish comprises a menu and a menu is offered every day. There is at least one dish in a menu. A menu is described by the menu name, day offered, and menu type (breakfast, lunch, dinner or snacks). Menu name and dish name are unique.

2. A virologist is assigned with a set of virus to study. A virologist studies a certain set of viruses in a laboratory. A laboratory is used by four (4) virologists only. There are many viruses stored and studied under a laboratory. A laboratory must be used by at least one virologist. A virologist is described by a virologist id, name, birthday, age, years in service and virologist rank which can be derived from the years of service. The virologist id identifies the virologist.

3. A news writer writes one or more news articles for a newspaper. A news article can be written by more than one news writer. A news article has a news title, news body, a byline, one or more pictures and an erratum (if any). A news writer has a news writer name which is unique, department to which he or she belongs to (e.g., sports, regional), social networking accounts (twitter, facebook, instagram, etc.) and an email address.