

# SOFTWARE ENGINEERING



# DATABASE MANAGEMENT SYSTEMS

**ENTITY RELATIONSHIP DIAGRAMS (ER DIAGRAMS)**

## Lesson 02 – Entity Relationship Diagrams (ER Diagrams)

### Entity Relationship Model

ER model is a high level logical data model which can be used in designing a database.

The ER model uses 3 main constructs:

#### Entities:

An “entity” is an object with a physical existence.

Eg: Person, Car, Employee, Student

There are two types of Entities.

#### 1. Strong entity

It has its own key attribute (primary key).

Eg: Empno can take as a key attribute for the employee entity. Therefore, employee is a strong entity. Indexno can take as a key attribute for the student entity. Therefore, student is a strong entity.

#### 2. Weak entity

Entity types that do not have key attributes of their own are called weak entities. Consider the Dependents of an Employee. Without an Employee Dependent cannot exist. Therefore, Dependent is a Weak Entity.

#### Attributes:

Each entity has properties called “Attributes” that describes the entity.

A Student has attributes like: Name, NIC, Index\_No, Address

A Course has attributes like: Name, Cno, Duration

Types of Attributes

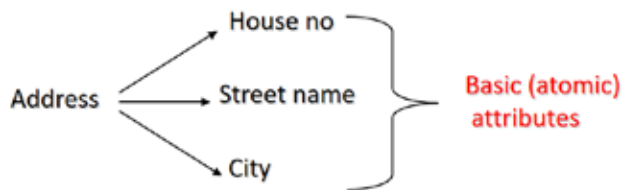
#### 1. Simple or Atomic Attributes

An attribute that is not divisible (broken down into smaller components).

Eg: Salary

#### 2. Composite Attributes

Composite attributes can divide into more basic attributes.



### 3. Single valued & multi valued attributes

Most attributes have a single value.

Eg: Salary

Some attributes can have more than one value.

Eg: “Location” attribute of a Department

“College Degree” attribute of a Person

“Qualification” attribute of a Lecturer

### 4. Stored attributes

An attribute whose value is fixed and cannot change.

Eg: Date of Birth, NIC

### 5. Derived attributes

An attribute whose value can be calculated from a stored attribute.

Eg: Age can be calculate using DOB

## Relationships:

A relationship is an association (connection) between two or more entities.

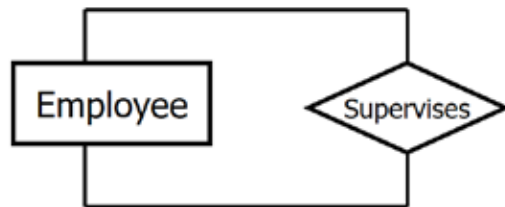
Relationship Degree

The degree of a relationship type is the number of participating entities in a given relationship.

### 1. Unary

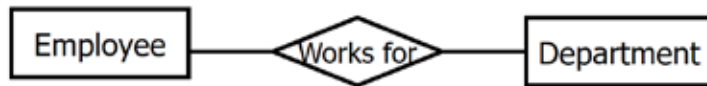
An Entity has a relationship to itself.

Eg. Supervisor/Employee



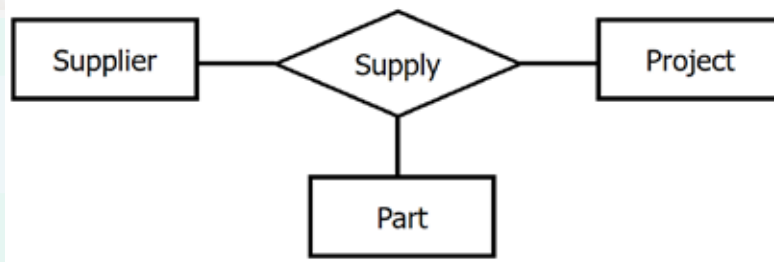
### 2. Binary

In a binary relationship, two entities are connected.












### 3. Ternary

For a particular relationship three entities are connected.



## Basic Symbols in ER Diagram

Symbol	Meaning
	Entity type
	Weak entity type
	Relationship type
	Identifying relationship type
	Attribute
	Key attribute
	Multi-valued attribute
	Composite attribute
	Derived attribute

*Figure 2.0.1 Basic Symbols of a ER Diagram*

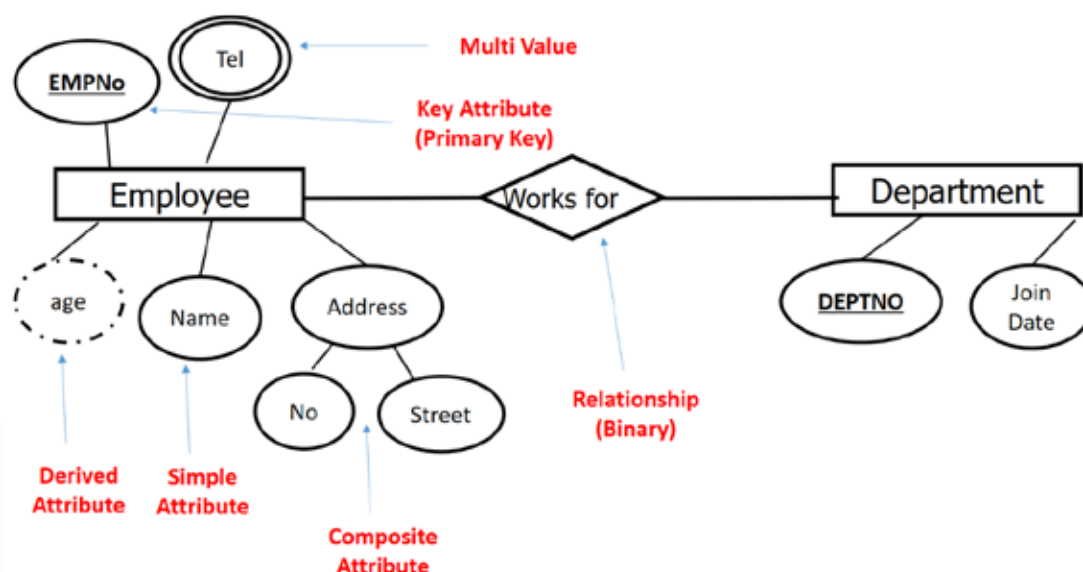


Figure 2.0.2 Simple Example for ER Diagram

## Example 01

A library Database is used to store information about books, publishers and members of the library. The Database keeps track of the following data:

For each book

BookID (unique), Title

For each publisher

Publisher name (unique), Address (publisher may have several office branches),  
Phone\_no

For each library member

Member card no (unique), Name, Address, Phone no

Consider the following Relationships:

The library books are borrowed by the members. The books are published by the publishers.

Draw an ER diagram indicating the attributes and relationships.

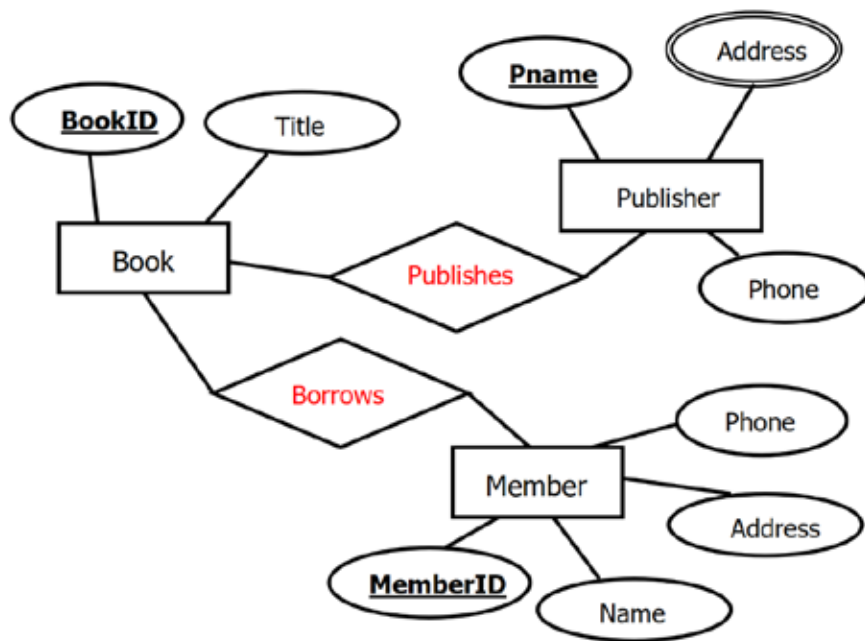


Figure 2.0.3 ER Diagram Example 01

## Example 02

A Book Publishing Company publishes books on various subjects. The books are written by authors and edited by the Editors of the Company. To maintain the standards of its books, the Company has assigned experienced authors to supervise the junior authors. The Company sells its books to book sellers. The database will keep track of the following data:

For each Book

Book Number (unique), Book Name, Edition, Year Published.

For each Author

Author Number (unique), Author Name, Address, Telephone Number Company Joined Date.

For each Editor

Editor Number (unique), Editor Name, Address, Telephone Number, Subject Area.

For each Seller

Company Name (Unique), Shipping Address (Many), Contact person Name, Telephone Number and Credit Limit.

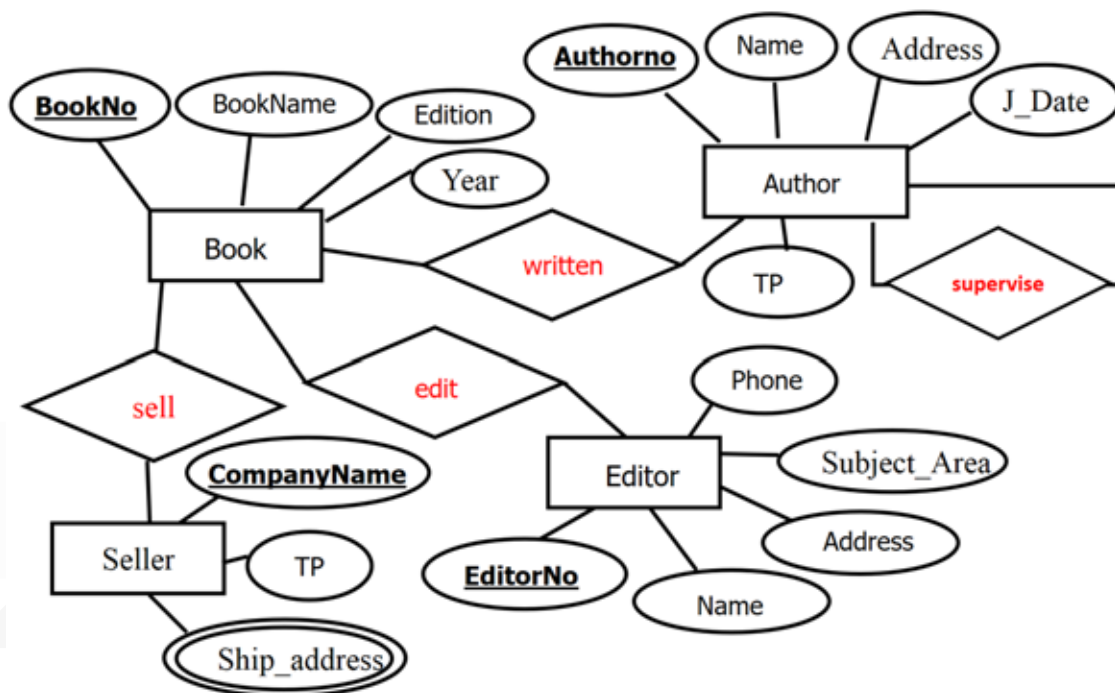


Figure 2.0.4 ER Diagram Example 02

### Important Facts to Remember

- Do not use Plural for the Entities
- Entity always should be a Noun
- If the attribute has two words, do not keep space between those two words. Instead of space you can use underscore ( \_ )
- Relationship should be present tense or present perfect tense. Do not use past tense

### Relationship Constraints

A relationship usually has certain constraints that limits the possible combinations of entities that may participate in relationship instance. There are two types of relationship constraints.

1. **Cardinality ratio**
2. **Participation**

## Cardinality Ratio

The cardinality ratio specifies the number of relationship instances that an entity can participate.

Common cardinality ratios for binary relationship types are:

1. One-to-one (1:1)
2. One-to-many (1:m)
3. Many-to-many (m:n)

### One-to-One (1:1)

An entity in A is associated with at most one entity in B, and an entity in B is associated with at most one entity in A.

Eg. A department entity to the employee who manages that department.

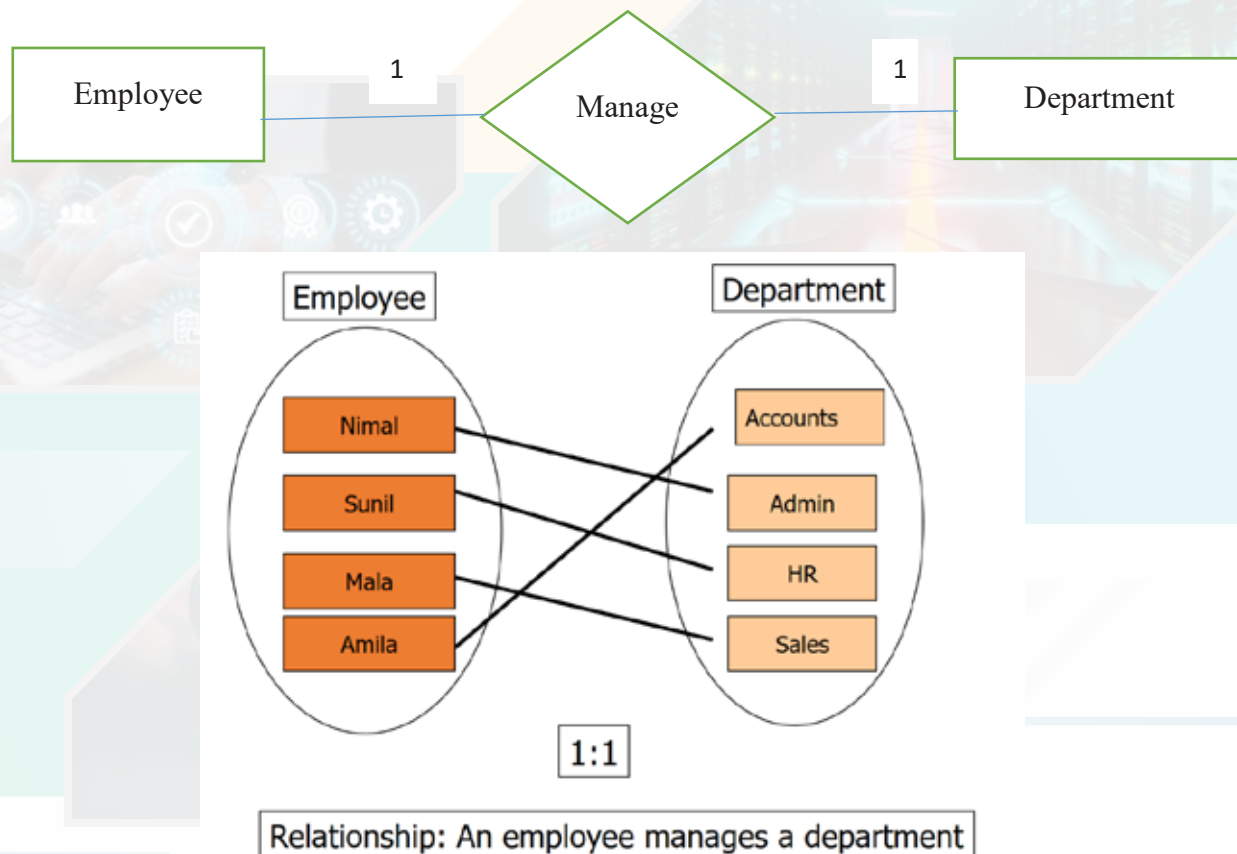


Figure 2.0.5 One to One Relationship Example

### One-to-many (1:m)

An entity in A is associated with a number of entities in B. An entity in B, however can be associated with at most one entity in A.

Eg. One employee can work in one department but in one department there can be many employees.

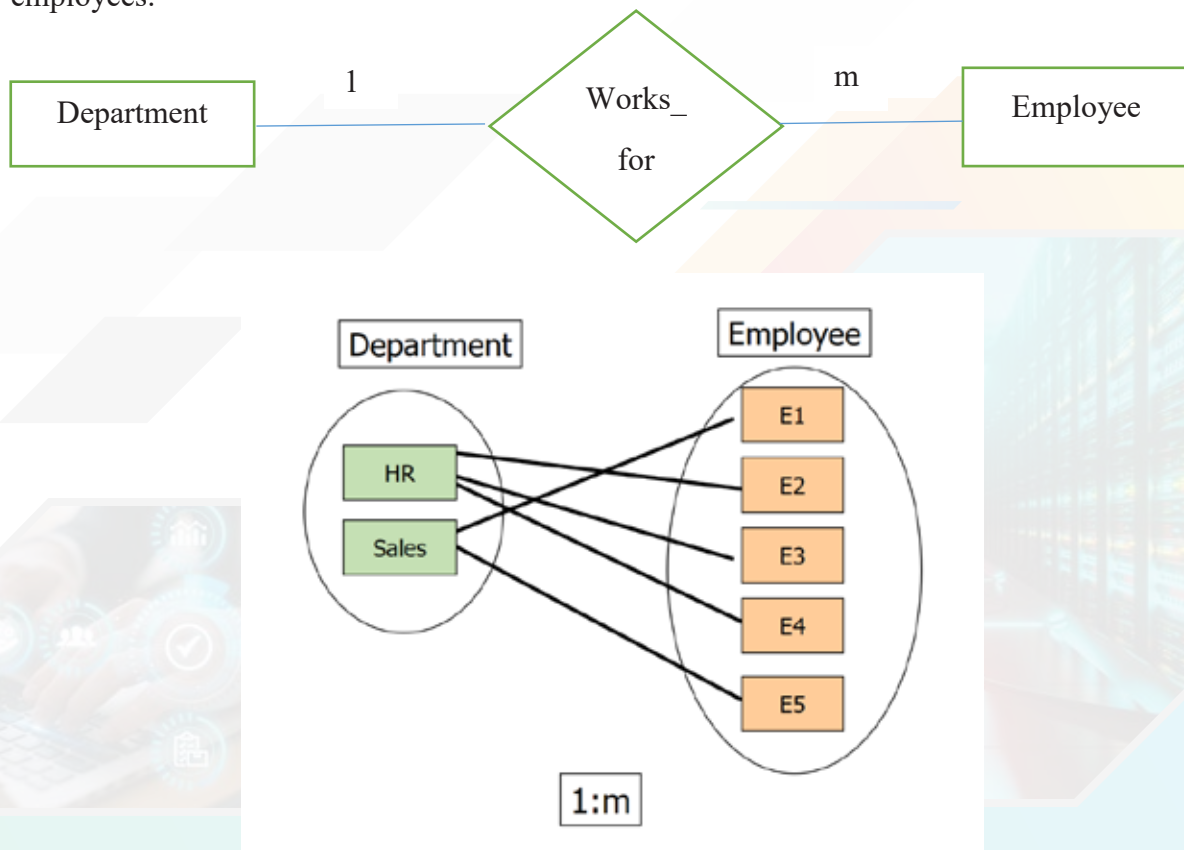
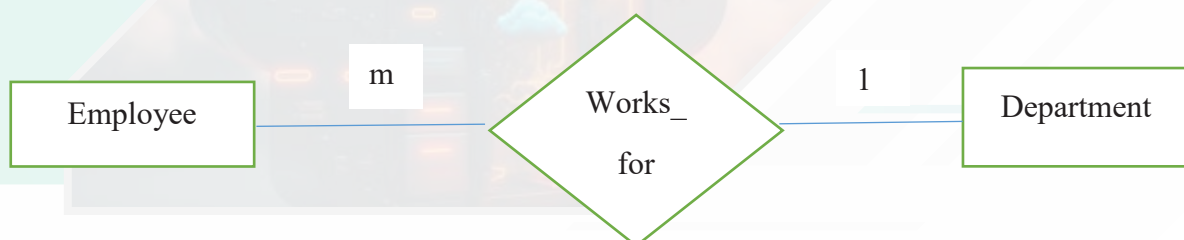


Figure 2.0.6 One to Many Relationship Example

If you change the Entities, you will get m:1 relationship. But it is same as the above one. Only difference is the way you read the relationship.



### Many-to-many (m:n)

An entity in A is associated with a number of entities in B, and an entity in B can be associated with any number of entities in A.

Eg. If one employee can work in several projects and one project has several employees, then the cardinality ratio is m:n.

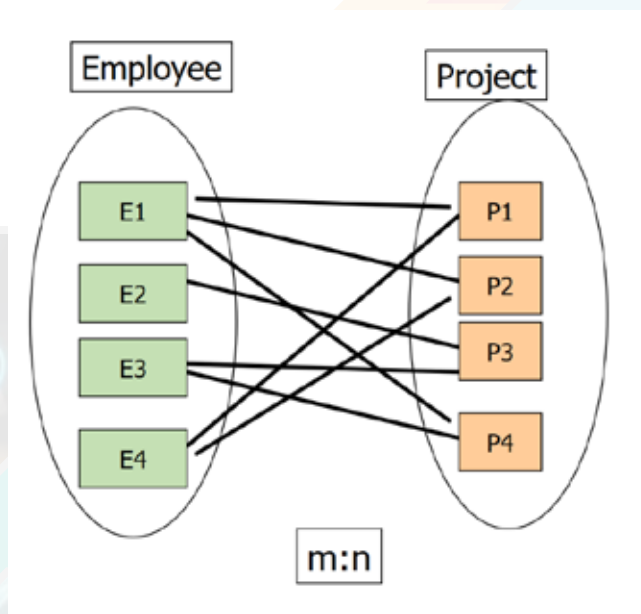


Figure 2.0.7 Many to Many Relationship Example

Still you have doubts? Refer Tutorial Guide and do Simple Exercises for Cardinality ratios



## Participation Constraint

It specifies whether the existence of an entity depends on its being related to another entity via the relationship type.

There are two types of participation constraints:

1. Total participation
2. Partial participation

### Total participation

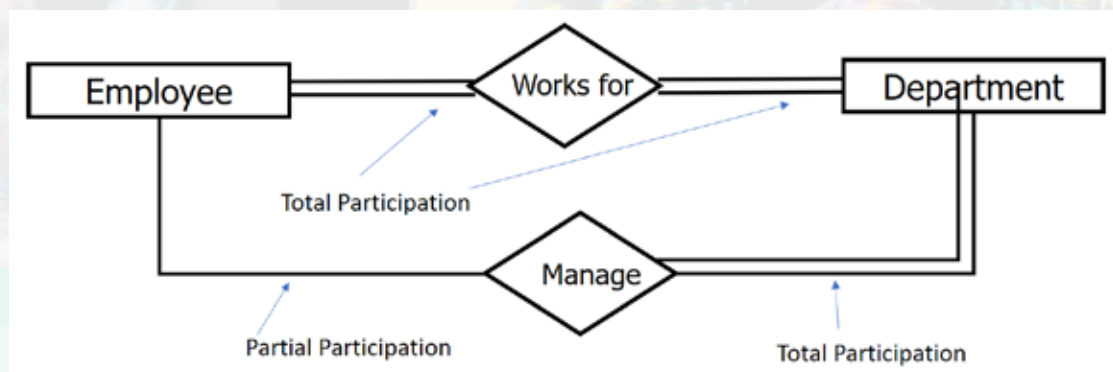
Every entity in this set must be related to another entity.

Eg: If a company policy states that every employee must work in a department, then an employee entity can exist only if it participates in a relationship with the Department entity.

### Partial participation

It means that only some or part of the set is related.

Eg: The relationship “manages” between the entities Employee and Department is partial as only one employee manages a department and not all.



Does all the employees work for a Department:

- If it is Yes, then it is a Total Participation

Does all the departments have employees working for it:

- If it is Yes, then it is a Total Participation

Does all the Employees manage a Department:

- If it is No, then it is a Partial Participation

Does all the departments have an Employee managing it:

- If it is Yes, then it is a Total Participation

## ER Diagram Exercise 01

A database used in an order-entry system is to contain information about customers, items, and orders. A customer can place many orders but an order is placed by only one customer. Orders consists of many items and the same item can be placed in many orders. The DB keeps track of the following data:

For each Customer

Customer number (unique), "Ship to" address (several per customer), Balance, Credit limit, Discount.

For each Order

Order no, "Ship to" address, date of order.

For each Item

Item number (unique), Item Description, Unit Price, Stock Quantity

The DB also keeps track of the Order quantity and quantity outstanding of each item for a given order.

Draw an ER diagram and mentioned all your assumptions.

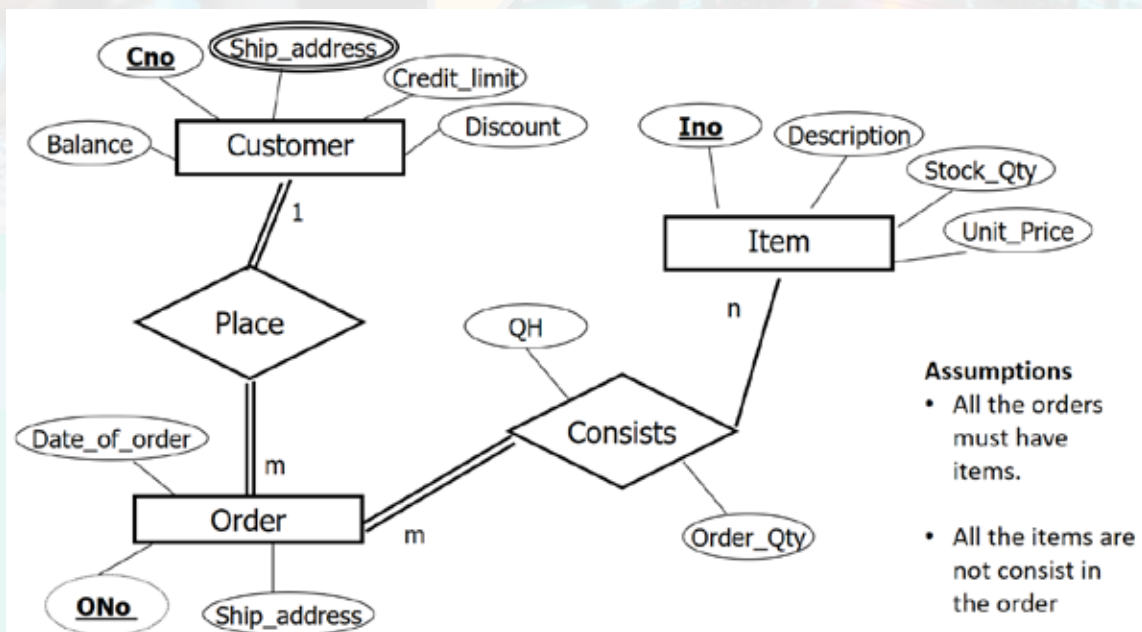


Figure 2.0.8 ER Diagram Exercise 01

### Assumptions

- All the customers are placing orders.
- All the orders must have a customer.
- All the orders must contain items.
- All the items are not consisting in the order.

## ER Diagram Exercise 02

The ABC University has several departments. Each department is assigned several lecturers and a lecturer may be assigned to more than one department. A department offers several courses, but a course is offered by only one department. A lecturer teaches in a number of courses, but a lecturer may be on leave and may not teach any course or be assigned to a department. Each course may be taught by more than one lecturer. The university DB keeps track of the department number (unique), department name, lecturer name, lecturer id (unique), course names (unique), number of subjects and duration of the course.

Draw an ER diagram indicating the attributes and relationships.

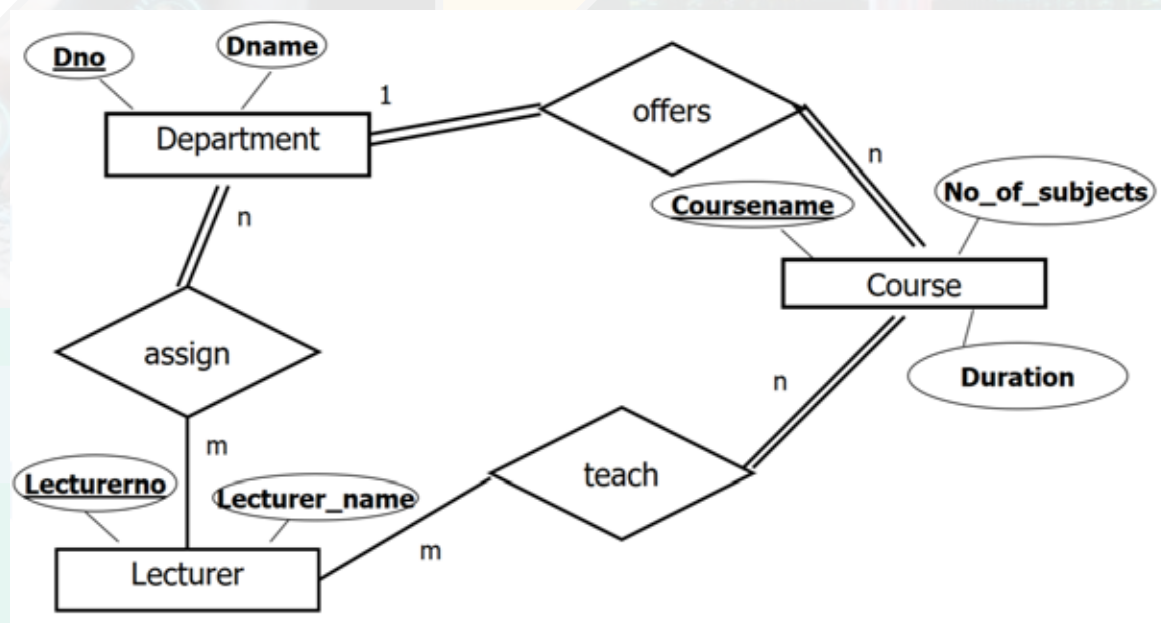


Figure 2.0.9 ER Diagram Exercise 02

### Assumptions

- All departments must have lecturer/s.
- All lecturers are not assigned to a department.
- All courses must have a lecturer/s but all lecturers are not teaching in the course.
- All department/s must offer course/s.

### Important Facts to Remember

- *Since we are not gathering actual requirements from the user You must write Assumptions.*
- *Always first do the Cardinality Ratio then do the Participation Constraint. When you do the Participation Constraint do not think about Cardinality Ratio.*

### ER Diagram Exercise 03

A General hospital consist of a number of wards. Each ward has a unique ward number, a name and the number of patients in that ward. A doctor is assigned to a single ward but a ward can have many doctors.

A ward hosts a number of patients. Each patient's admission number (unique), name, address and telephone number and date of admission is stored.

The hospital DB stores the prescribed treatment for a patient. Each treatment has a unique treatment number and description. It also keeps track of the treatment dosage for each patient. One patient may be prescribed more than one treatment and the same treatment can be prescribed to many patients.

The hospital also keeps track of the hospital ID number of each doctor (unique), name, address, telephone number and specialization of each doctor. A patient is assigned to one doctor, but a doctor can treat many patients.

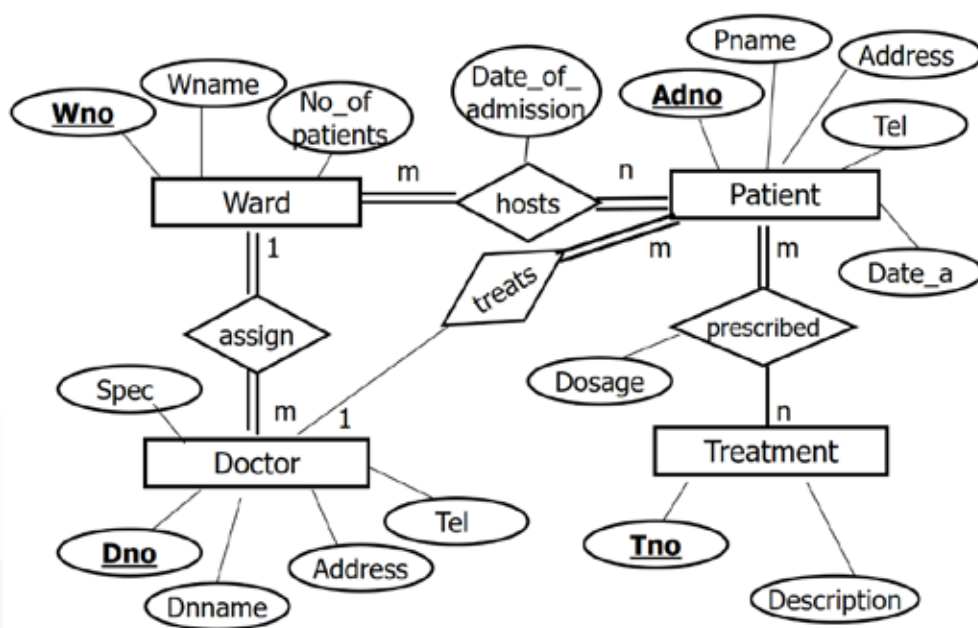


Figure 2.0.10 ER Diagram Exercise 03

#### Assumptions

- One Patient can be assign to many Wards.
- All the wards must have doctor and all the doctors assigned to a ward.
- All the patients assigned to doctors but all the doctors do not treat to patient.
- All the treatments won't prescribe to a patient.
- All the patients are hosted in a ward.

### ER Diagram Exercise 04

The company is organized into departments. Each department has a name, a unique number, telephone number. Each department is managed by one employee and a department will have only one employee managing it. The DB will keep track of the start date when that employee began managing the department. A department may have several locations. A department controls a number of projects, each of which has a name a unique number and a single location.

The DB will store each employees name, unique employee number, address, salary and DOB. An employee is assigned to one department but may work on several projects which are not necessarily controlled by the same department. The DB will keep track of the number of hours per week that an employee works on each project. Each employee is supervised by a supervisor.

The DB will keep track of the dependents of each employee for insurance purposes. It will store each dependent's name, DOB, and relationship to the employee.

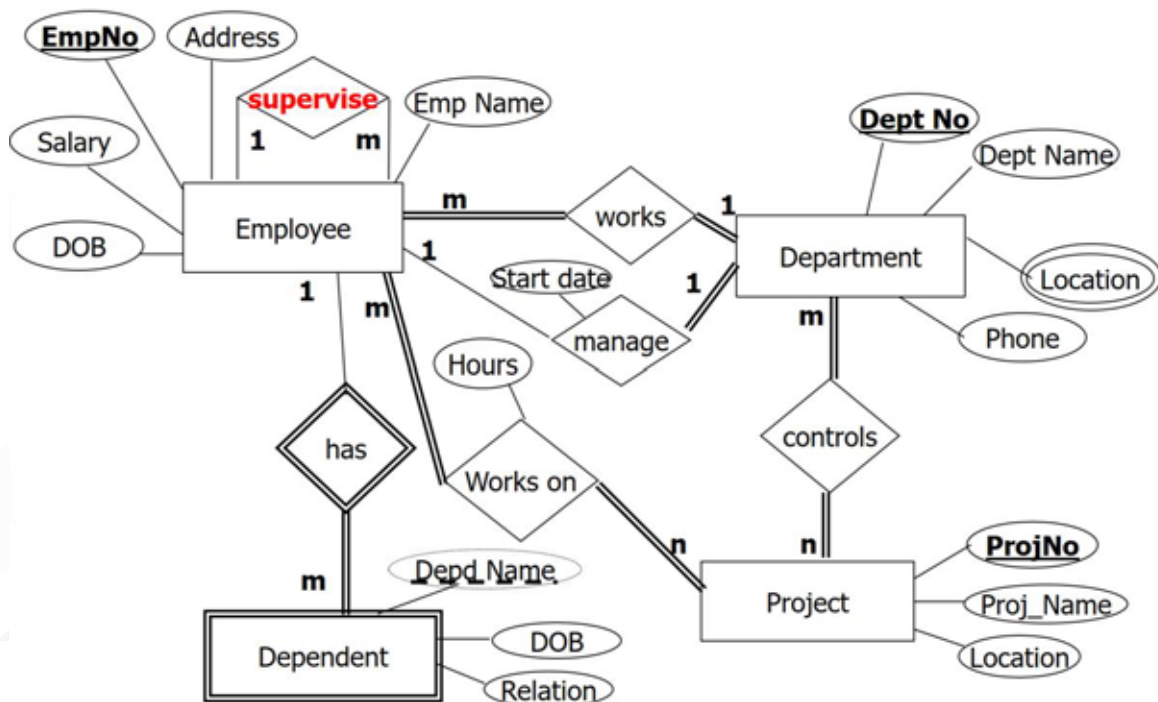


Figure 2.0.11 ER Diagram Exercise 04

#### Assumptions

- All the employees are working under a department and all departments have employees.
- All the departments have a manager but all the employees are not a manager.
- All the employees are not supervised by a supervisor.
- All the departments control a project and all the projects have a department.
- All the employees are working on a project and all the projects have employees.
- All the dependents must have an employee but all the employees don't have dependent.

Still wondering  
about ER Diagrams?  
Refer Tutorial Guide  
and do ER Exercises

