

Objetivo del ejercicio:

1. *Aprender las Relaciones Reflexivas.*
2. *Aprender el uso del predicado NULL.*
3. *Aprender el concepto de un SELF JOIN.*

Tipos de relaciones según el grado en el Modelo Entidad – Relación:

Según el número de entidades que se asocian en una relación podemos tener diferentes tipos de relaciones según el grado:

- **Binarias:** Relación que asocia dos entidades.
- **N-arias:** Relación que asocia N entidades (siendo N mayor que 2) (estas relaciones no se dan en el modelo relacional)
- **Reflexivas:** Relaciona una entidad consigo misma.

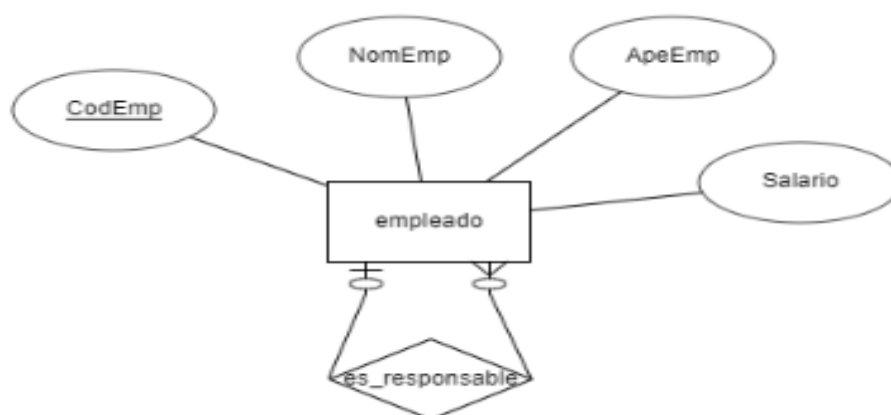
El modelo relacional solo contempla las relaciones reflexivas y binarias.

Se desea almacenar información de los empleados de una pequeña empresa. La información que se debe guardar de cada empleado es su código, nombre, apellidos y salario. Además, queremos conocer los empleados que son responsables (o enseñan) a otros empleados. Un empleado puede no ser responsable de ningún otro empleado, o bien, serlo de muchos otros. Un empleado puede no tener a ningún otro empleado que se responsabilice de él o bien tener un solo responsable.

Se pide:

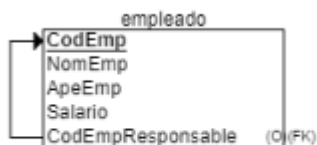
1. Modelar la base de datos. Para ello haremos:

- Diseño Conceptual de Datos utilizando un Diagrama o Modelo Entidad-Relación. Lo hacemos en papel y lo pasamos a la Herramienta CASE ERD Plus.

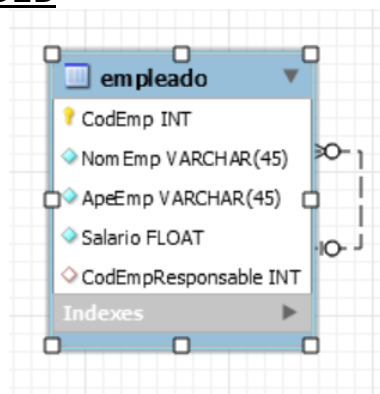


- b. Diseño Lógico de Datos utilizando un Diagrama de Estructura de datos (DED). Lo hacemos en papel y lo pasamos a la Herramienta CASE MySql Workbench. En este apartado también vamos a poner el Diagrama Referencial que genera ERD Plus a partir del Modelo Entidad-Relación. Recuerda que el Diseño Lógico de Datos es hacer el modelo relacional y para ello podemos hacer un DED o un Diagrama Referencial.

Diagrama Referencial



DED



- c. Diseño Físico de Datos. Creamos la base de datos y las tablas en SQL. Utiliza el autoincremento para la PK.

```
CREATE TABLE empleado
(
  CodEmp INT NOT NULL AUTO_INCREMENT,
  NomEmp VARCHAR(50) NOT NULL,
  ApeEmp VARCHAR(80) NOT NULL,
  Salario DECIMAL(6,2) NOT NULL,
  CodEmpResponsable INT,
  PRIMARY KEY (CodEmp),
  FOREIGN KEY (CodEmpResponsable) REFERENCES empleado(CodEmp)
);
```

2. Insertar datos desde phpmyadmin utilizando la sentencia INSERT INTO del LMD de SQL. Teniendo en cuenta que los empleados que son responsable de otros, no se responsabilizan ningún otro de los mismos. Es decir, los empleados responsables, tienen la FK a NULL. Uno de los empleados responsables debe llamarse José Pérez Pérez

```
INSERT INTO empleado (NomEmp,ApeEmp,Salario, CodEmpResponsable)
VALUES ('José', 'Pérez Pérez',2500,NULL),
       ('Ana', 'López López',2500,NULL),
       ('Juan', 'Jiménez Pérez',1500,1),
       ('Rosa', 'Martín Martín', 1500,1),
       ('Ramón', 'González González', 1500,2);
```

3. Realizar las siguientes consultas:

- Obtener los nombres y apellidos de los empleados responsables.

```
SELECT NomEmp, ApeEmp
FROM empleado
WHERE CodEmpResponsable IS NULL;
```

- Obtener de cada empleado, el nombre y apellidos de su responsable.

```
SELECT emp.NomEmp,emp.ApeEmp,resp.NomEmp AS 'Nombre Responsable',resp.ApeEmp as 'Apellidos Responsable'
FROM empleado emp LEFT JOIN empleado resp ON (emp.CodEmpResponsable=resp.CodEmp);
```

- Obtener los nombres y apellidos de los empleados de los cuales es responsable el empleado José Pérez Pérez.

```
-- Primero vemos la información
SELECT *
FROM empleado E JOIN empleado Responsable ON (E.CodEmpResponsable=Responsable.CodEmp);

-- Ahora obtenemos lo que queremos
SELECT *
FROM empleado E JOIN empleado Responsable ON (E.CodEmpResponsable=Responsable.CodEmp)
WHERE Responsable.NomEmp='José' AND Responsable.ApeEmp='Pérez Pérez';

SELECT E.NomEmp,E.ApeEmp
FROM empleado E JOIN empleado Responsable ON (E.CodEmpResponsable=Responsable.CodEmp)
WHERE Responsable.NomEmp='José' AND Responsable.ApeEmp='Pérez Pérez';
```