

Tarea 1: Modelo E-R extendido

Carlos Tonatihu Barrera Pérez
Profesor: Hernández Contreras Euler
Bases de Datos
Grupo: 2CM1

13 de marzo de 2017

Índice

1. Introducción	2
2. Desarrollo	3
2.1. Subclases/Superclases	3
2.2. Especialización/Generalización	3
2.3. Herencia	5
2.4. Categorías	5
2.5. Notación y herramientas CASE	6
3. Conclusión	8
Referencias	8

1. Introducción

En el diseño de bases de datos existe el modelo entidad-relación que representa la estructura lógica global de la base de datos mediante el uso de conceptos básicos como las entidades, las relaciones y los atributos, estos conceptos y sus derivados nos brindan una gran ayuda en el diseño de la base de datos.[1]

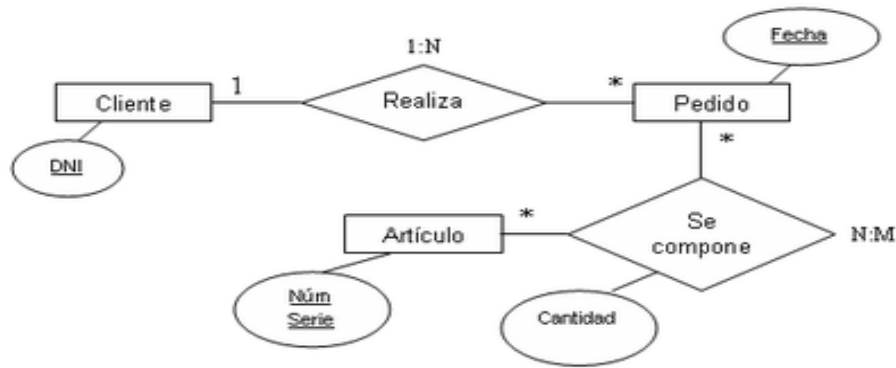


Figura 1: Ejemplo de un diagrama entidad-relación

A pesar de que el modelo Entidad-Relación nos permite modelar la mayoría de las características de las bases de datos existen casos en que este modelo no es suficiente por lo que se tiene que utilizar una versión extendida de este llamado modelo E-R extendido.

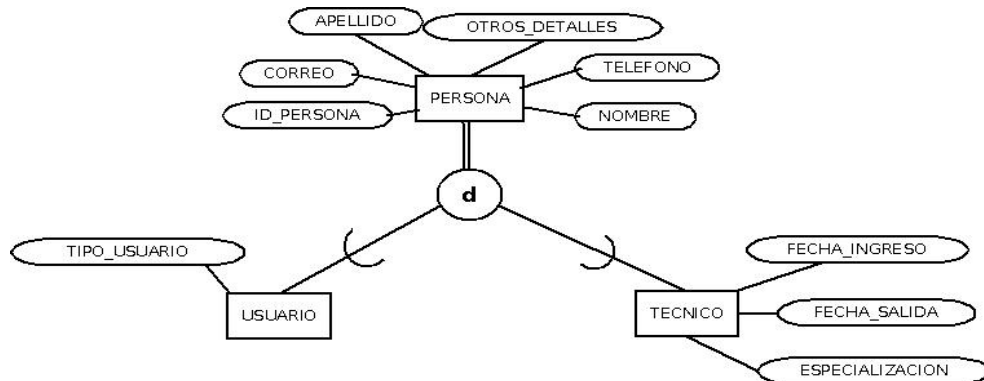


Figura 2: Ejemplo de un diagrama entidad-relación extendido

Este modelo sera explicado en la parte de desarrollo donde se utilizan ejemplos para ilustrar sus principales características ademas de que se presentan los elementos que conforman su notación y algunas herramientas case para poder diseñarlos.

2. Desarrollo

El modelo entidad-relación extendido (ERE o EER por sus siglas en inglés) es un modelo de datos conceptual capaz de describir la estructura (y funcionalidad) de bases de datos de una forma fácil y directa de entender mediante una notación gráfica.[2] Además, el modelo ERE incluye todos los conceptos del entidad-relación e incorpora nuevos conceptos los cuales son: **subclases/superclase**, **especialización/generalización**, **herencia** y **categorías**.

2.1. Subclases/Superclases

Esta característica se presenta cuando las entidades de los modelos tienen subgrupos de entidades que deben ser representados en el modelo debido a su importancia.[3] Un ejemplo que ilustra esto se encuentra en la *figura 3* en donde vehículo es la superclase (o entidad de nivel superior) y coche y camión son las subclases (o entidades de nivel inferior).

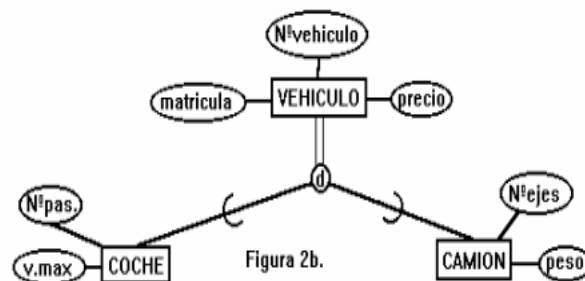


Figura 3: Un vehículo debe de ser forzosamente un coche o un camión pero no ambos.

Es importante mencionar que una entidad no solo puede ser miembro de una subclase sino que también debe de ser miembro de la superclase. Además, una entidad puede ser miembro de varias subclases.

2.2. Especialización/Generalización

Especialización es el proceso de definir un conjunto de subclases de una superclase. El conjunto de subclases está basado en algunas características distintivas de las entidades en la superclase.

La generalización es el proceso reverso a la especialización. Varias clases con características en común son generalizadas en una superclase.[3]

Tanto la especificación como la generalización presentan las siguientes restricciones.

- Definidas por predicado (predicate-defined) o por condición (condition-defined). Usada cuando podemos determinar aquellas entidades que se convertirán en miembros de cada subclase mediante una condición.
- Definidas por atributo (attribute-defined).

- Definidas por atributo. Si todas las subclases en una especialización tienen una condición de pertenencia en el mismo atributo de la superclase.
- Definida por el usuario (user-defined). Si no existe condición para determinar la pertenencia a una subclase. Además la clasificación se hará manualmente.
- Restricción de disyunción (disjointness constraint). Puede ser de dos tipos.
 - Disjunta (disjoint). La entidad puede ser miembro de una única subclase de la especialización.
 - Solapada (overlapping). Una misma entidad puede ser miembro de una o más subclases de una especialización
- Restricción de completitud (completeness constraint). Existen dos tipos.
 - Total especifica que cada entidad en la superclase debe ser miembro de alguna subclase.
 - Parcial permite a las entidades no pertenecer a ninguna subclase.

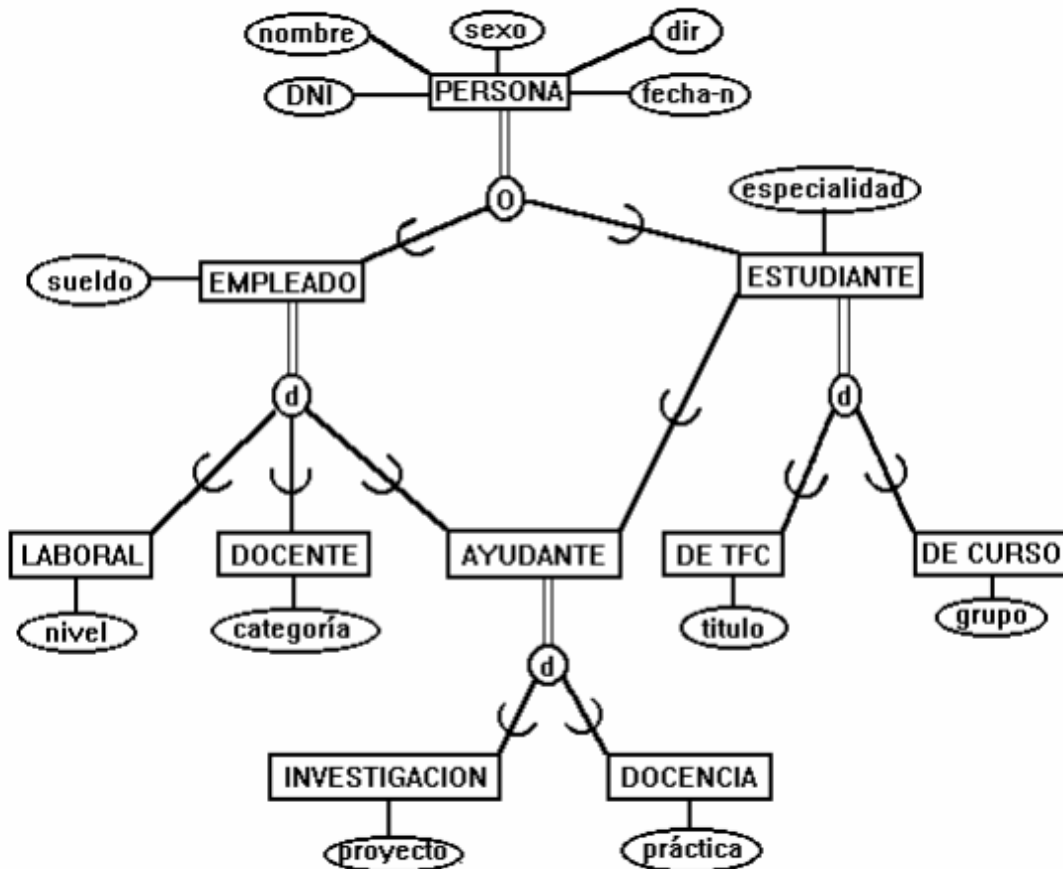


Figura 4: Ejemplo con la mayoría de los elementos de la generalización y la especialización

2.3. Herencia

Una propiedad de las subclases y superclases creadas mediante la especialización y la generalización es la herencia de atributos. Esto hace referencia a que los atributos de las entidades de nivel superior son heredados por las entidades de nivel inferior esto no ocurre en sentido inverso.[2]

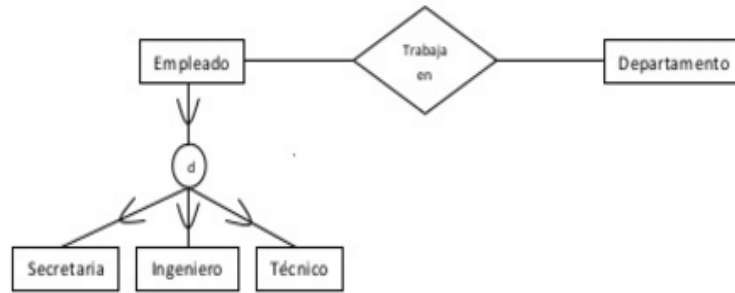


Figura 5: Ejemplo de herencia simple.

También las subclases heredan la participación en los conjuntos de relaciones en los que participa su superclase. Además la herencia se divide en simple y múltiple en donde la múltiple hace referencia a que una subclase compartida puede tener varias superclases como se presenta en la *figura 5* en este caso se dice que se tiene un retículo, malla o red (lattice), en la herencia simple una subclase tiene solo una superclase un ejemplo de esto se puede observar en la *figura 6* y es básicamente una estructura de árbol.[3]

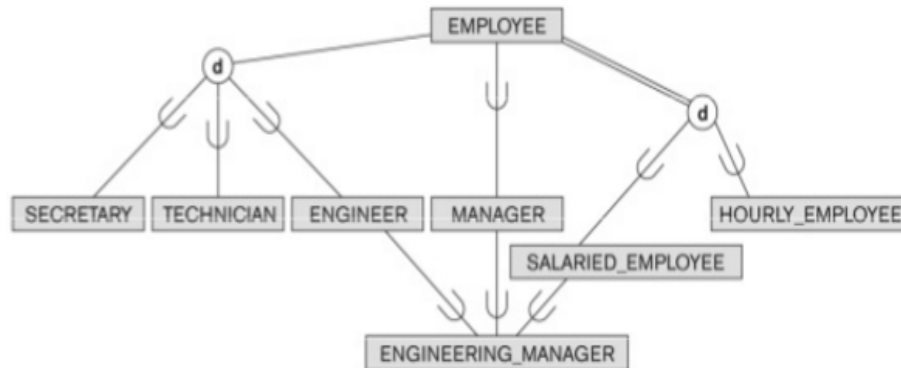


Figura 6: En este caso engineering_manager tiene herencia múltiple de engineer, manager y salaried_employee.

2.4. Categorías

Un tipo de unión o categoría es una subclase que representa una colección de objetos que son un subconjunto de la unión de distintos tipos de entidad, las categorías tienen las siguientes características.[1]

- Siempre tienen dos o mas superclases.
- Parcial permite a las entidades no pertenecer a ninguna subclase.
- Participación en una categoría Existen dos tipos.
 - Total. Si todas las superclases de la categoría deben ser miembros de la categoría. Esto también puede modelarse como una generalización disjunta.
 - Parcial. Si no todas las superclases deben ser miembros de una categoría

2.5. Notación y herramientas CASE

Los diagramas ERE heredan los elementos que conforman el diseño de los diagramas ER pero agregan nuevos componentes para representar sus características de herencia, superclases/subclases, especialización/generalización y tipo de unión.

Existen varias formas de representar los elementos de este tipo de diagramas pero una muy utilizada es la siguiente. En la relación entre superclases y subclases se utiliza el símbolo de pertenencia en la línea.

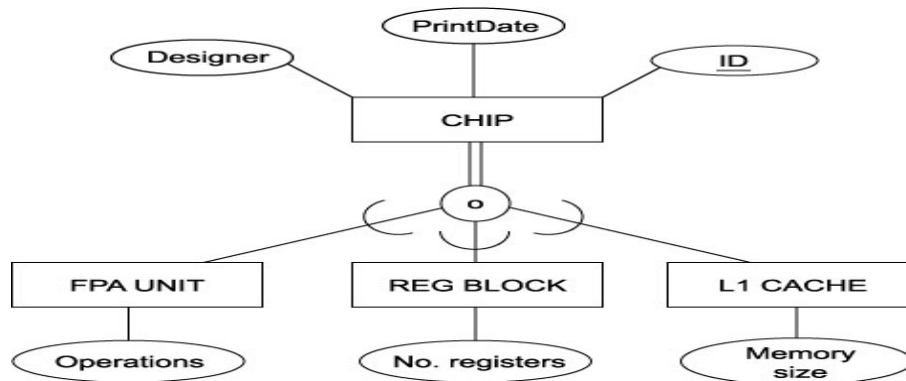


Figura 7: Ejemplo de herencia simple.

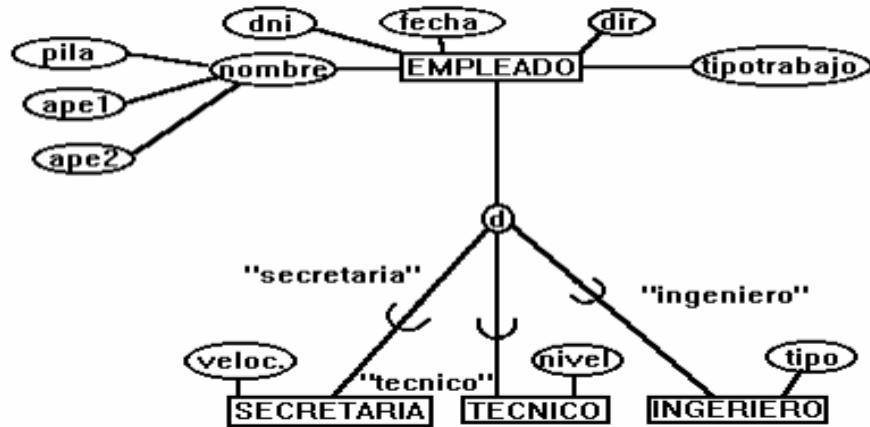


Figura 8: Ejemplo de herencia simple.

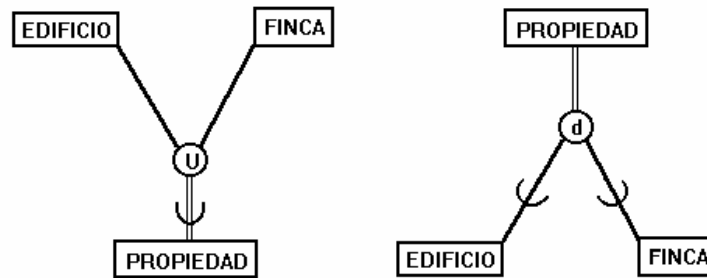


Figura 9: Ejemplo de herencia simple.

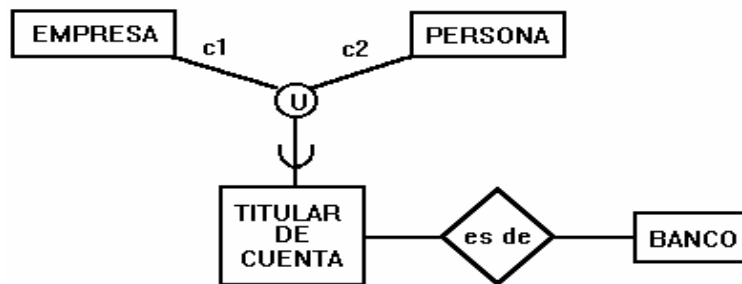


Figura 10: Ejemplo de herencia simple.

Para el desarrollo de este tipo de diagramas existen herramientas CASE que tienen la opción de utilizar elementos de este tipo de diagramas o en su defecto se utilizan diagramas de clases para representar este modelo. Algunas de estas herramientas son.

- Umbrello

- StarUML
- EERCASE
- DB Designer

3. Conclusión

Los diagramas entidad-relación extendidos son mucho mucho mas completos que los ER por lo que nos ofrecen mas herramientas para del diseño de nuestra base de datos pero debido a su complejidad hay que conocerlos a fondo y los elementos que los conforman para poder utilizarlos de la manera más optima posible ya que de lo contrario en vez de generar un beneficio pueden llegar a ser perjudiciales en el desarrollo de un proyecto.

Otro punto importante a considerar es en que momento utilizarlo debido a que por su nivel de complejidad es bueno para proyectos que demanden mayor detalle así que en sistemas que necesiten una base de datos pequeña no tan compleja seria una mejor idea utilizar el modelo entidad-relación básico.

Referencias

- [1] A. Silberschatz, *Fundamentos de Diseño de Bases de Datos*. McGraw-Hill, 2007.
- [2] J. G. Moreno, “Modelo Entidad-Relación Extendido.” <https://es.slideshare.net/gerardomo/modelo-de-entidad-relacin-extendido>, 2013. [Consultado: 2017-03-12].
- [3] R. A. Elmasri, *Fundamentals of database systems*. Pearson, 2011.