

TAREA 4 - Tipos de entidades

Vences Santillán Carlos Eduardo

31 de agosto de 2025

Dependencia e independencia de existencia

Definición: La *dependencia de existencia* ocurre cuando la existencia de una entidad está subordinada a la existencia de otra entidad, mientras que la *independencia de existencia* ocurre cuando una entidad puede existir sin la necesidad de estar asociada a otra.

Explicación: En una base de datos, las entidades son como los objetos o cosas del mundo real que queremos representar: clientes, facturas, productos, etc. Algunas entidades pueden existir por sí mismas, sin necesidad de relacionarse con otras; a esto se le llama *independencia de existencia*. Por ejemplo, un cliente puede registrarse en la base de datos aunque todavía no haya comprado nada, porque su información personal (nombre, dirección, teléfono) tiene sentido por sí misma.

En cambio, hay entidades que solo tienen sentido si están relacionadas con otra, y a eso se le llama *dependencia de existencia*. Por ejemplo, un “detalle de factura” (los productos específicos que se incluyen en una factura) no puede existir sin la factura que lo contiene. Si tratáramos de registrar un detalle de factura sin una factura asociada, esa información quedaría incompleta o sin sentido. Por eso, decimos que el detalle depende de la existencia de la factura.

Entidad débil

Definición: Una entidad débil es aquella que no posee una clave primaria propia y depende de una entidad fuerte para su identificación, utilizando la clave primaria de la entidad fuerte junto con un discriminador parcial para formar su clave primaria compuesta.

Explicación: Una entidad débil es aquella que, por sí sola, no tiene un atributo único que permita identificarla en la base de datos. Recordemos que normalmente a cada entidad se le asigna una clave primaria, que es un dato o conjunto de datos que la diferencia de todas las demás (por ejemplo, un número de cliente o un número de producto). Pero en el caso de las entidades débiles, no es posible definir una clave primaria sin apoyarse en otra entidad.

Imaginemos un sistema de facturación. Tenemos la entidad **Factura**, con su número único (ejemplo: “Factura 001”). Dentro de cada factura hay varios productos listados con sus cantidades y precios, lo que se llama **DetalleFactura**. Este detalle no tiene sentido por sí mismo, porque “Detalle 1” podría repetirse en varias facturas. Para identificarlo de manera única, necesita unirse al número de factura a la que pertenece. Así, el detalle se identifica por la combinación de **NúmeroFactura** + **NúmeroDetalle**, lo que lo hace una entidad débil.

Dependencia de identificación

Definición: La *dependencia de identificación* ocurre cuando una entidad débil se identifica a través de la clave primaria de una entidad fuerte, formando con ella una clave compuesta que asegura la unicidad de sus registros.

Explicación: La dependencia de identificación se refiere a que una entidad débil necesita obligatoriamente usar la clave primaria de otra entidad (llamada entidad fuerte) para poder ser identificada de manera única. Es decir, no basta con sus propios atributos; necesita heredar parte de la identificación de otra entidad. Por eso, la relación entre una entidad débil y su entidad fuerte es mucho más estrecha que en otros casos: la entidad débil simplemente no puede “vivir” sin esa relación.

En el ejemplo de **Factura** y **DetalleFactura**, cada detalle solo puede identificarse si conocemos el número de la factura. “Detalle 1” por sí mismo no significa nada, pero “Factura 001 – Detalle 1” sí es único y válido. En el modelo entidad-relación, esto se representa con símbolos especiales (rectángulos dobles para las entidades débiles y rombos dobles para la relación), para mostrar que la existencia y la identificación del detalle dependen completamente de la factura.

Referencias

- [1] Elmasri, R., & Navathe, S. B. (2015). *Fundamentals of Database Systems* (7th ed.). Pearson.
- [2] Harrington, J. L. (2016). *Relational Database Design and Implementation* (4th ed.). Morgan Kaufmann.
- [3] Kroenke, D. M., & Auer, D. (2017). *Database Concepts* (8th ed.). Pearson.
- [4] García-Molina, H., Ullman, J. D., & Widom, J. (2009). *Database Systems: The Complete Book* (2nd ed.). Pearson.
- [5] Connolly, T. M., & Begg, C. E. (2015). *Database Systems: A Practical Approach to Design, Implementation, and Management* (6th ed.). Pearson.