Modelo Relacional para la Gestión de Datos en Ciencia de Datos

Autor
Martín Vladimir Alonso Sierra Galvis

Gestión de Datos, Maestría en Ciencia de Datos
Pontificia Universidad Javeriana Cali

Versión 2.0
Santiago de Cali, febrero de 2023

Tabla de contenido

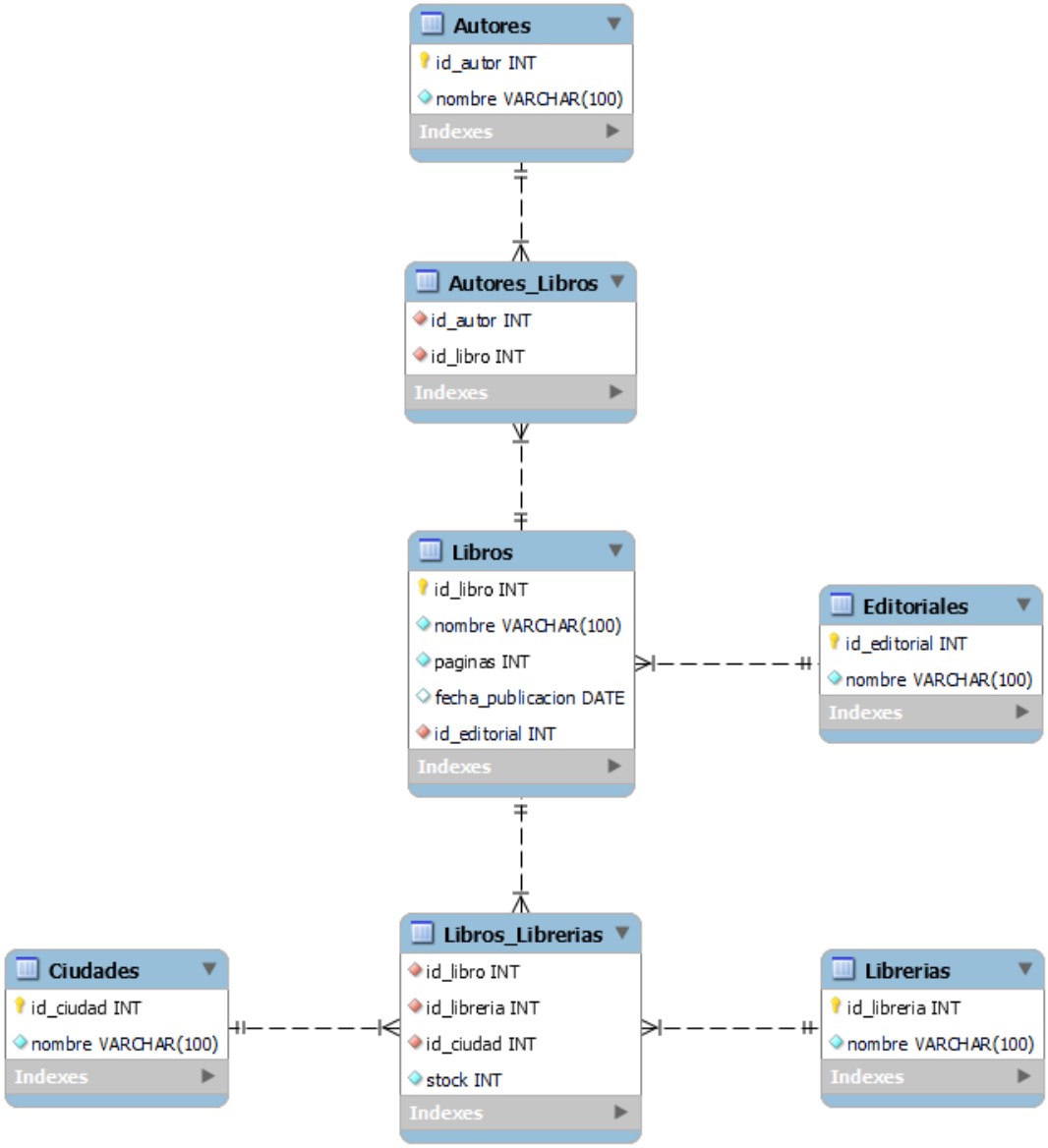
El Modelo Relacional

¿Qué es un modelo?

El término modelo tiene gran variedad de significados dependiendo del área de conocimiento en el que se use. De manera general, en el área de la tecnología y la información, un modelo es una representación abstracta de un sistema real que nos va a permitir entender su estructura y comportamiento. Cuando hablamos de bases de datos, un modelo nos permitirá conocer, de forma conceptual o lógica, la estructura y organización de los datos en una base de datos.

¿Qué es un Modelo Relacional?

Un Modelo Relacional es un modelo lógico que permite representar la base de datos relacional por medio de las tablas que la conforman, sus respectivos campos y las relaciones que existen entre dichas tablas. Es una herramienta muy utilizada por Ingenieros de Datos, para construir un diseño de la base de datos que, posteriormente, les sirva de orientación para realizar la implementación física de la misma en un Sistema Gestor de Base de Datos Relacional. Esto porque dichos profesionales son dueños o tienen gobernanza sobre los datos de la base de datos. Por otro lado, los Científicos de Datos, no somos dueños de los datos. Nuestra tarea, no es crear bases de datos; **nuestra tarea consiste en consultar los datos**, **ya almacenados en una base de datos**, **con el fin de procesarlos y realizar analítica o aplicar diversos algoritmos sobre ellos**, (p ej.: Machine Learning). Por esta razón, el Modelo Relacional, más que una herramienta de diseño, será para nosotros una guía que nos mostrará cómo se encuentran organizados los datos en la base de datos, algo que facilitará mucho los procesos de consulta. La siguiente imagen muestra un ejemplo de un modelo relacional.

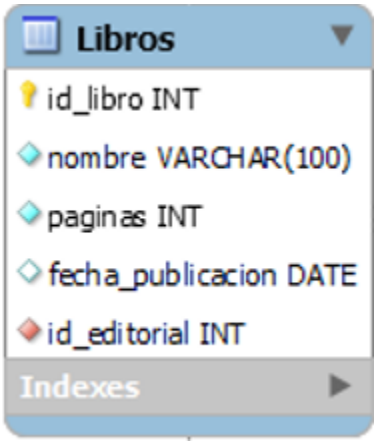


Este modelo representa una base de datos relacional de una aplicación de libros. Los datos están organizados y relacionados de tal forma que se puede almacenar los nombres de los libros, su número de páginas y su fecha de publicación; los nombres de los autores de los diferentes libros registrados; las editoriales que se encargaron de la edición y publicación de cada libro; y las librerías y ciudades donde se encuentran disponibles los libros. A continuación, estudiaremos algunos conceptos importantes de las bases de datos relacionales mientras se explica más a detalle las diferentes partes del modelo.

Conceptos importantes

Tabla

Las tablas son el núcleo de una base de datos relacional, la estructura en la que se almacenan los datos de una forma organizada. Las tablas se componen de dos elementos principales: las filas y las columnas. Las filas representan cada uno de los registros que se van almacenando en la base de datos. Por su parte, las columnas representan los atributos que contiene cada registro. En cada columna se registran los valores que va a tener cada registro en sus respectivos atributos. Tomemos como ejemplo la siguiente imagen del Modelo Relacional.



Este pequeño recuadro, cuyo nombre es **Libros**, es la representación lógica de la tabla Libros. Como podemos observar, la tabla contiene cinco atributos: id_libro, nombre, paginas, fecha_publicacion y id_editorial. Esto quiere decir que la tabla tiene cinco columnas donde se registran los valores de estos atributos. En el Sistema Gestor de la Base de Datos Relacional, la tabla se vería más o menos así.

id_libro	nombre	paginas	fecha_publicacion	id_editorial
1	Juego de Tronos	600	01/0!/98	2
2	Segunda Fundación	270	01/01/53	3

Esta tabla contiene dos libros registrados: Juego de Tronos y Segunda Fundación. Para cada uno de ellos se registran un identificador, su nombre,

el número de páginas y su fecha de publicación. Aquí podremos notar algo interesante: la editorial está representada por un número en lugar de un nombre. A lo largo del documento iremos explicando la razón de esta organización.

Tipo de dato

Es importante resaltar que cada atributo en una tabla posee lo que se llama un tipo de dato. Un tipo de dato, básicamente, es la representación del dominio de un valor. Por ejemplo, podemos encontrar datos que pertenecen al dominio numérico como números naturales, números enteros o números decimales; también podemos encontrar datos que pertenecen al dominio de texto, como cadenas de texto o caracteres. Los tipos de datos definen qué procesos o manipulaciones podemos realizar a los datos con los que estamos interactuando. En el caso de nuestro ejemplo, los atributos id_libro, paginas y id_editorial tienen asignado el tipo de dato INT, es decir, sus valores deben ser números enteros; por otro lado, el atributo nombre es de tipo VARCHAR2(100), lo que quiere decir que sus valores son textos de máximo 100 caracteres; finalmente el tipo de dato del atributo fecha_publicacion es DATE, por lo que sus valores deben ser fechas. Este concepto de tipos de datos tiene una gran relevancia en la Ciencia de Datos, así que es muy importante tenerlo siempre en cuenta cuando trabajamos con los datos.

Llave primaria (PK)

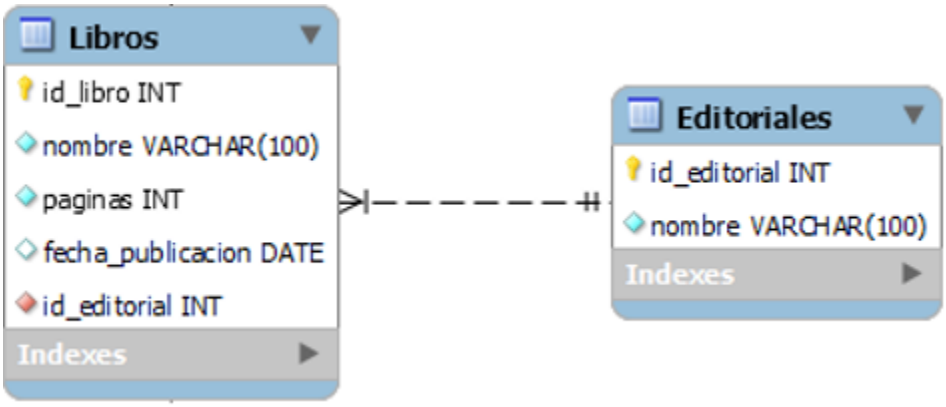
Si observamos con más detalle la representación lógica de la tabla Libros, notaremos que el atributo id_libro está acompañado por un ícono particular: una llave dorada. Esto significa que dicho atributo es una llave o clave primaria. Las llaves primarias son uno de los elementos más importantes de las bases de datos relacionales. Tienen como finalidad otorgar un identificador único a cada registro de una tabla; hace las veces de "documento de identidad" del registro. Adicionalmente, nos otorgan un mecanismo para evitar la redundancia de datos, pues cada registro, al tener una llave primaria, debería considerarse como único, es decir, no debería haber dos registros exactamente iguales en una tabla. Por último, y relacionado con la reducción de redundancia, son el elemento que permite implementar relaciones entre datos de diferentes tablas, funcionalidad que otorga su nombre a este tipo de bases de datos.

Llave foránea (FK)

Recordemos que en el apartado **Tabla**, habíamos visto que las editoriales que publicaron los libros aparecían con un número en lugar de un nombre, pero, ¿por qué? Resulta que dichos números corresponden a **llaves primarias** que se encuentran en la tabla Editoriales. Esto es lo que se conoce como una llave foránea, un identificador que referencia un registro específico, ubicado en una tabla distinta. Las llaves foráneas son el segundo elemento que permite la implementación de las relaciones y lo hacen bajo un concepto llamado Normalización que veremos más adelante en este documento.

Relación

Pilar de las bases de datos relacionales; precisamente de ahí viene su nombre. Las relaciones, en esencia, son vínculos o asociaciones entre los datos de las distintas tablas que conforman la base de datos. Tomemos nuevamente como ejemplo la tabla Libros. Sabemos que aquí existe una relación con la tabla Editoriales, esto porque en Libros tenemos la columna id_editorial con llaves foráneas, mismas que referencian a las llaves primarias en Editoriales. En el Modelo Relacional podemos ver dicha la relación.



Las tablas Libros y Editoriales se encuentran unidas por una línea, lo que indica una relación entre ellas. Por convención, normalmente los profesionales encargados del diseño de una base de datos le colocan el mismo nombre tanto al campo que representa la llave foránea como al campo que representa la llave primaria en la otra tabla. Lo anterior lo podemos evidenciar en el nombre id_editorial. Ahora bien, la posición de la

llave foránea es importante si deseamos entender bien qué significa la relación que estamos viendo. Volvamos a los datos en la tabla que teníamos antes.

id_libro	nombre	paginas	fecha_publicacion	id_editorial
1	Juego de Tronos	600	01/0!/98	2
2	Segunda Fundación	270	01/01/53	3

En este caso podemos identificar que un libro se relaciona únicamente con una editorial ya que en el campo id_editorial tenemos permitido colocar solamente un única llave foránea por registro. Pero, lo anterior no quiere decir que yo no pueda repetir el valor de una llave foránea en diferentes registros. Por ejemplo, para un tercer libro, de nombre Tormenta de Espadas, el id_editorial podría volver a ser 2. Esto porque hace parte de la misma saga a la que pertenece Juego de Tronos, Canción de Hielo y Fuego, así que es normal que tengan la misma editorial. Desde el punto de vista de las editoriales se puede decir que una editorial tiene la facultad de editar y publicar varios libros. Lo que acabamos de describir se conoce como Cardinalidad.

Cardinalidad

La cardinalidad define el número de registros vinculados entre las tablas que configuran una relación. En una base de datos podemos encontrar tres tipos principales de cardinalidades.

Cardinalidad 1:1

Esta cardinalidad es la más rara de las tres. Nos indica que para dos tablas relacionadas A y B, los registros de la tabla A se relacionan cada uno con un único registro de la tabla B, y los registros de la tabla B se relacionan con un único registro de la tabla A. En este caso la llave foránea puede ir en cualquiera de las dos tablas de la relación.

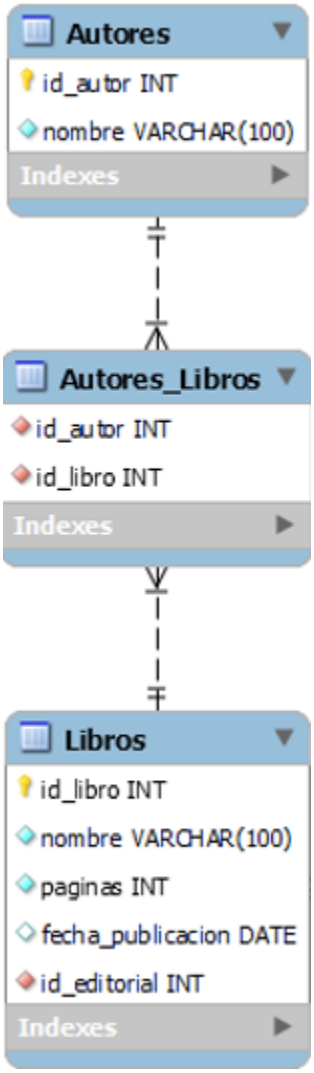
Cardinalidad 1:N

Esta cardinalidad nos indica que para dos tablas relacionadas A y B, los registros de la tabla A se relacionan con varios registros de la tabla B, mientras que los registros de la tabla B se relacionan con un único registro de la tabla B. En este caso, la llave primaria de los registros de la tabla A se

convierte en llave foránea en los registros de la tabla B. En nuestro Modelo Relacional ya vimos este tipo de cardinalidad en la relación entre Libros y Editoriales.

Cardinalidad N:M

También llamada cardinalidad muchos a muchos, nos indica que para dos tablas relacionadas A y B, los registros de la tabla A se relacionan con varios registros de la tabla B, y los registros de la tabla B se relacionan con varios registros de la tabla A. Este tipo de cardinalidad se puede reconocer por la presencia de una tabla intermedia C entre las tablas A y B. En Nuestro Modelo Relacional tenemos como ejemplo la relación entre las tablas Autores y Libros, misma que se muestra a continuación.



Aquí la tabla C sería Autores_Libros. Esta tabla intermedia tiene la responsabilidad de contener llaves foráneas que referencian a las dos tablas que hacen parte de la relación. Para el ejemplo, el significado es: un autor puede escribir uno o más libros y un libro puede ser escrito por uno o varios autores.

Reglas de Normalización

La Normalización es un conjunto de reglas, comúnmente llamadas formas, que se aplican al momento de diseñar e implementar una base de datos relacional. Estas reglas existen, básicamente, para evitar posibles problemas de redundancia de datos, para que los datos sean consistentes y para que la base de datos sea fácilmente moldeable. Ahora bien, las reglas de Normalización no es un tema en el que vayamos a profundizar en este curso puesto que, como acabamos de ver, estás reglas están más orientadas a la etapa de diseño y creación: quienes deben tener mayor conocimiento de ellas son profesionales como Desarrolladores e Ingenieros de Software o Ingenieros de Datos. Como Científicos de Datos, para lo único que nos interesaría conocer las reglas de Normalización es para entender el por qué de la estructura del Modelo Relacional, el por qué los datos están organizados de tal manera en la base de datos relacional y para realizar procesos de desnormalización de datos.

Las Formas Normales son cinco, pero las primeras tres son las más importantes y las que más nos vamos a encontrar en situaciones reales. Comúnmente se les conoce como 1FN, 2FN y 3FN.

1FN

Una base de datos que cumpla con la primera forma normal tiene una llave única para todos los registros de todas las tablas, es decir, hace uso de llaves primarias. Además los datos almacenados deben ser **atómicos**, o lo que es lo mismo, no deben tener la posibilidad de dividirse en otros datos. Finalmente, las tablas no deben ser propensas a que sus columnas estén variando continuamente (p. ej.: agregar nuevas columnas). Si esto sucede, las columnas variantes deberían convertirse en registros en una nueva tabla.

2FN

Una base de datos en segunda forma normal debe cumplir con la primera forma normal y además verificar que todos los atributos o columnas de una

tabla dependan de la llave primaria de la tabla. La mejor forma de detectar que una columna no depende de la llave primaria es identificar que sus valores se repiten. Cuando esto pasa, los valores de dicho atributo deben pasar a ser registros en una nueva tabla y se debe crear otra tabla donde se relacionan los identificadores. Se utiliza para solucionar el problema de redundancia de datos en relaciones de cardinalidad N:M.

3FN

Una base de datos en tercera forma normal debe cumplir con la segunda forma normal y además debe eliminar de toda tabla los atributos que no dependan de la llave primaria y convertirlos en registros de otras tablas diferentes. Se diferencia de la segunda forma normal en que esta se utiliza para solucionar redundancia de datos en relaciones de cardinalidad 1:N.

Estas tres formas normales son consideradas las mejores prácticas para la creación de bases de datos relacionales, lo que a la larga nos ayudará a nosotros a acceder a la información de una manera más organizada y legible. El grave problema de la Normalización aparece cuando vamos a consultar datos para realizar analítica de datos. Debido a la organización de los datos, este tipo de consultas tienen un rendimiento muy bajo lo que ocasiona que las respuestas tarden mucho tiempo. Por esta razón, en algún momento, tendremos que aplicar técnicas para desnormalizar datos y para ello debemos saber qué está normalizado y qué no.

Para terminar y a modo de reflexión, el Modelo Relacional que revisamos en este documento se encuentra en 3FN.

Referencias

1. DataCademia. *Relational Data Modeling - Cardinality*. [En línea]: disponible en https://datacadamia.com/data/type/relation/modeling/cardinality#articles_related. [Accedido en 2023].
2. Google Cloud. *¿Qué es una base de datos relacional?* [En línea]: disponible en https://cloud.google.com/learn/what-is-a-relational-database?hl=es. [Accedido en 2023].
3. hdeleon.net. *Normalización de BASE de DATOS*. Julio de 2022. [En línea]: disponible en https://www.youtube.com/watch?v=fxbC4cwnb1U. [Accedido en 2023].
4. IBM. *Cardinality*. Marzo de 2021. [En línea]: Disponible en https://www.ibm.com/docs/en/cognos-analytics/10.2.2?topic=relationships-cardinality. [Accedido en 2023].
5. Indeed Editorial Team. *What is a Relationship in Database? (Definition and Types)*. Julio de 2022. [En línea]: disponible en https://in.indeed.com/career-advice/career-development/what-is-relationship-in-database. [Accedido en 2022].
6. Oracle. *¿Qué es una base de datos relacional (sistema de gestión de bases de datos relacionales)?* [En línea]: disponible en https://www.oracle.com/co/database/what-is-a-relational-database/. [Accedido en 2023].