

AUTORENTA

SANTIAGO CENTENO CASTAÑO

P3

PEDRO FELIPE GÓMEZ BONILLA

**CAMPUSLANDS
APOLO
RUTA BACKEND
FLORIDABLANCA
2024**

Tabla de Contenidos

Introducción	4
Caso de Estudio	5
Planificación	5
Construcción del Modelo Conceptual	5
Descripción	5
Gráfica	5
Descripción Técnica	6
Construcción del Modelo Lógico	6
Descripción	6
Gráfica	6
Descripción Técnica	6
Normalización del Modelo Lógico	6
Primera Forma Normal (1FN)	7
Descripción	7
Gráfica	7
Descripción Técnica	7
Segunda Forma Normal (2FN)	7
Descripción	8
Gráfica	8
Descripción Técnica	8
Tercera Forma Normal (3FN)	8
Descripción	8
Gráfica	8
Descripción Técnica	8
Construcción del Modelo Físico	9
Descripción	9
Código	9
Descripción Técnica	10
Diagrama E-R	10
Descripción	10
Gráfica	10
Descripción Técnica	10
Tablas	11
Descripción	11
Gráfica	11
Descripción Técnica	11
Relaciones entre Tablas	11
Descripción	11
Gráfica	12
Descripción Técnica	12
Inserción de Datos	12
Descripción	12

	Gráfica	12
	Descripción Técnica	12
Referencias		13

Introducción

En este documento me adentraré en el proyecto “Librería el mundo de Sofía”. Voy a analizar y documentar todo el proceso, para un entendimiento claro de porque ciertas decisiones fueron tomadas para satisfacer las necesidades del cliente. Esto se hará para lograr una base de datos limpia, robusta, fácil de modificar y entender para futuros analistas que vayan a ver y ampliar la base de datos.

Caso de Estudio

El proyecto “Librería el mundo de Sofía” presenta un sistema que requiere una base de datos para almacenar datos referentes a la compra y venta de libros. Tanto sus libros como sus transacciones tendrán detalles necesarios para un buen almacenamiento de la información, separando y asignando datos a diferentes tablas.

Planificación

Construcción del Modelo Conceptual

Para crear el modelo conceptual primero escribí en un cuaderno para organizar las entidades, sus atributos y sus relaciones. Esto me ayuda a poner la idea en la mesa y también es una buena práctica para mantener el orden. Después de hacer esto procedí a hacer el verdadero modelo conceptual el cual es digitalizado para que tenga una buena presentación y que sea de fácil entendimiento, tanto para el cliente como para los futuros analistas.

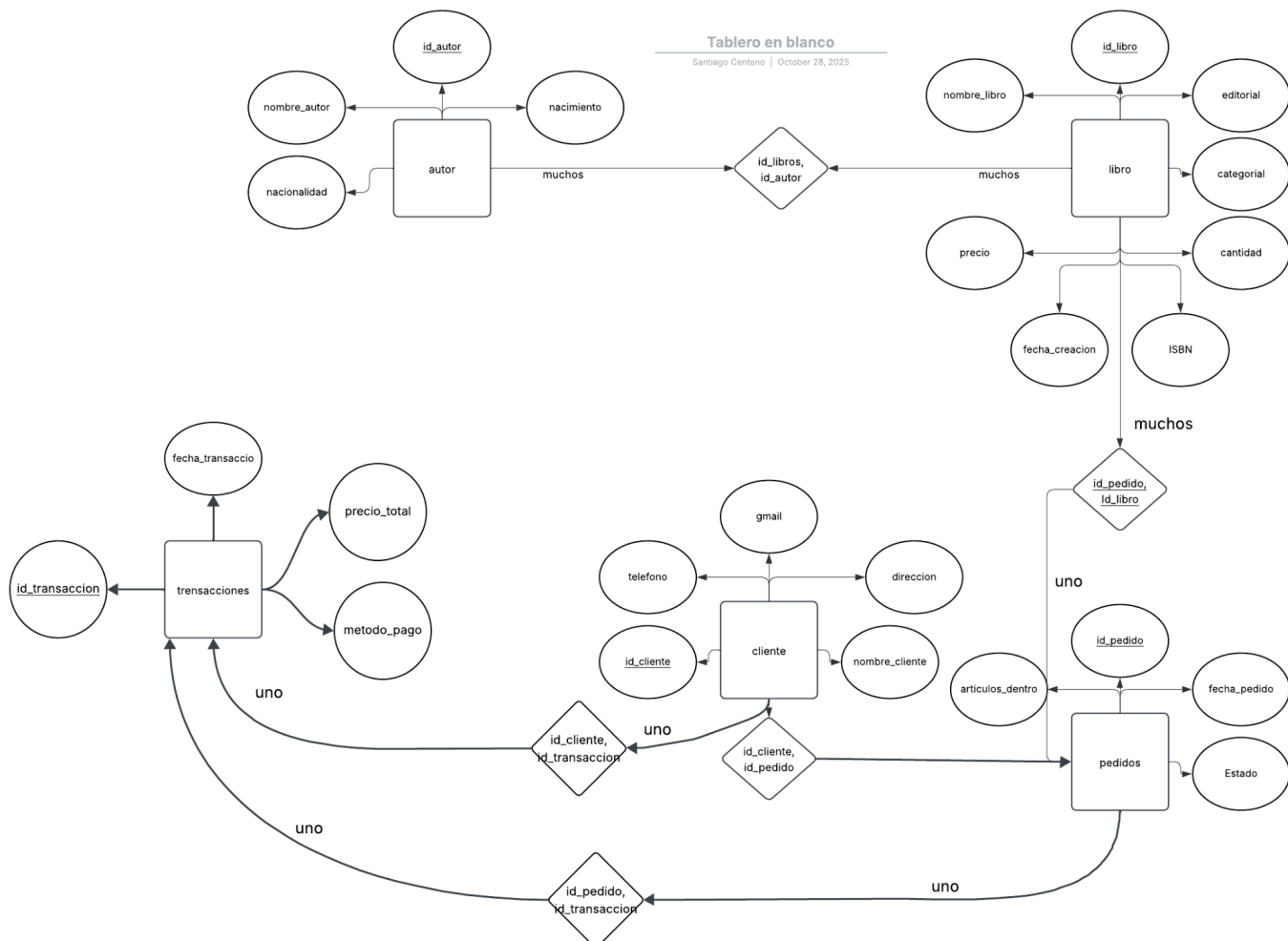
El modelo conceptual digitalizado contiene todas las tablas, su contenido y sus conexiones, todo esto de manera muy ordenada, teniendo en cuenta las formas normales 1, 2 y 3 que ayudan a mantener una base de datos limpia.

Descripción

El modelo conceptual tiene un total de 5 entidades y cada entidad tiene al menos 2 o más atributos. Las conexiones relacionan las tablas de manera lógica y siendo fiel a la segunda forma normal para darle más pureza a la base de datos. Este fue el primer paso para llegar a tener una idea clara de la base de datos. Se analiza y se separan entidades para evitar violar las reglas formales, dándole una forma mas logica a los datos, esto lo hace facil de entender para cualquier persona que lo mire, tambien es facil de explicar a aquellos que no estan adentrados en el mundo de base de datos.

El modelo conceptual es esencial en modelos de base de datos porque ayuda a entender la base de datos de manera superficial sin entrar en detalles técnicos. Agiliza el proceso de creación de los siguientes pasos como modelo lógico y modelo físico.

Gráfica



Descripción Técnica

Las entidades dentro del modelo conceptual presentan múltiples entidades, estas entidades de por sí tienen aún más atributos. Algunos atributos funcionan como llave primaria para dar una conexión entre atributos coherente. Cada atributo de la gráfica también tiene un orden correcto para que no parezca una sopa de letras sin sentido. Debido a su tamaño el modelo no tiene mucha complejidad de por sí, pero es importante no omitir ningún detalle para evitar errores lógicos. A continuación entraré en detalles con las entidades, sus atributos y conexiones lógicas dentro del modelo conceptual.

Entidades y atributos:

- **libros:** id libros (llave primaria), nombre, editorial, categoría, cantidad, precio, ISBN, fecha de creación.

- autor: id autor (llave primaria), nombre, fecha de nacimiento, nacionalidad.
- clientes: id clientes (llave primaria), dirección, número celular, email, nombre.
- pedidos: id pedidos, id clientes (llave foránea), id libros (llave foránea), cantidad, fecha de pedido.
- transacciones: id transacciones, id clientes (llave foránea), id pedidos (llave foránea), fecha de transacción, precio total, método de pago.

Cardinalidad:

Libros y Autor tienen una relación de muchos a muchos.

Pedidos y Libros tienen una relación de uno a muchos.

Pedidos y Transacciones tienen una relación de uno a uno.

Pedidos y Clientes tienen una relación de muchos a uno.

Transacciones y Clientes tienen una relación de uno a uno.

Construcción del Modelo Lógico

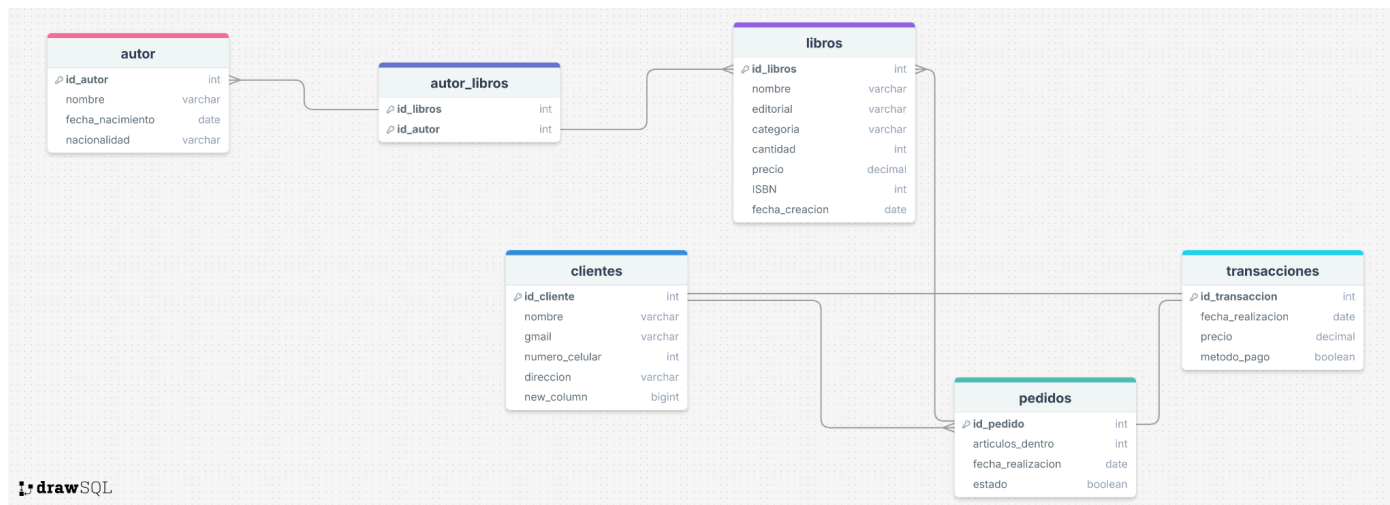
Para hacer el modelo lógico me base en gran parte por el modelo conceptual para hacer todas las relaciones, el único cambio relevante fue entre relaciones ya que dentro del modelo lógico algunas relaciones no son posibles sin antes implementar una tabla intermedia o tabla puente, como por ejemplo, para hacer una conexión de muchos a muchos entre la entidad autor y la entidad libros. Sin la tabla intermedia este proceso no hubiera sido posible en el modelo lógico ya que este también pone en marcha código funcional en sql que tiene límites lógicos que valen la pena mencionar.

Descripción

Este modelo lógico contiene la misma cantidad de tablas propuestas en el modelo conceptual, también tiene sus mismas relaciones con excepción de libros y autor ya que como fue mencionado anteriormente para esta relación se necesito una tabla intermedia que conectara ambas creando una relación de muchos a muchos. Añadir llaves primaria y foráneas para darle funcionalidad a las tablas. Es importante recordar que mientras se crea el modelo lógico también se genera automáticamente su código, al menos en esta página que está diseñada para eso.

El modelo lógico empieza a implementar código funcional para usar en SQL. El código presentado por la gráfica se genera automáticamente, aunque también es posible crear el código por sí mismo, se puede decir que es preferible copiarlo directamente del modelo lógico facilitando el proceso.

Gráfica



Descripción Técnica

Las tablas contienen los mismos atributos y las mismas conexiones que las del modelo conceptual. Se le añadieron llaves primarias y foráneas que ayudan a la lógica del código. Cada dato en cada tabla se diferencia por su tipo de dato, ya sea INT, decimal, varchar, booleano o date.

El modelo lógico se ve aún más ordenado que el modelo conceptual, pero más difícil de entender para personas que no estén adentradas en base de datos. Esta complejidad lo hace aún más completo para la creación del modelo físico, haciendo que crear el modelo físico sea solo codificar y listo.

Normalización del Modelo Lógico

En el proceso de creación del modelo lógico tuve en cuenta las 3 formas normales para tener una base de datos estructurada y utilizable en un entorno real. Estas formas normales se tuvieron en cuenta tanto para el inicio como para el final, porque es importante hacer una revisión estricta para que la normalización se cumpla de manera correcta.

Primera Forma Normal (1FN)

La primera forma normal indica que un dato solo puede contener un valor en la misma línea. Si un dato va a tener más de un valor se necesitan asignar varias líneas para este proceso.

Descripción

Me aseguré de que cada atributo tuviera su único valor. Para implementar la lógica de añadir nuevos atributos se necesita usar código, por lo tanto en el modelo lógico no tengo conocimiento para hacerlo.

Gráfica

libros	
id_libros	int
nombre	varchar
editorial	varchar
categoria	varchar
cantidad	int
precio	decimal
ISBN	int
fecha_creacion	date

Descripción Técnica

En la gráfica presentada se muestran datos con posibilidad de aumentar, estos datos como por ejemplo, un nuevo libro, se le implementara añadiendo un nuevo atributo a la tabla con una copia de los datos de la plantilla pero esta vez siendo asignados a un nuevo libro con una nueva id, etc...

Segunda Forma Normal (2FN)

La segunda forma normal se encarga de ordenar los datos en diferentes tablas. Una persona normal podría pensar que cliente y ciudad deberían ir juntos pero a veces en base de datos la separación de estos es esencial para mantener un orden y lógica correctos.

Descripción

Antes de hacer el modelo lógico e incluso antes de hacer el modelo conceptual esta forma normal fue utilizada para separar datos que personalmente creí que deben ir separados para no tener una base de datos saturada sino que funcional. La separación de tablas permite que se implemente y modifique lógica futura a la base de datos, como por ejemplo si en un futuro se necesita agregar una dependencia entre tablas. En este caso la segunda forma normal haría el acceso a cada tabla fácil para un programador de base de datos.

Gráfica

clientes	
id_cliente	int
nombre	varchar
gmail	varchar
numero_celular	int
direccion	varchar

transacciones	
id_transaccion	int
fecha_realizacion	date
precio	decimal
metodo_pago	boolean

Descripción Técnica

Teniendo tablas individuales con sus respectivas llaves se permite una futura utilización de las mismas, por ejemplo, si se va a añadir una tabla de gmail solo tendría que referenciar la llave primaria de clientes dentro de la tabla gmail, haciendo así una llave foránea útil para la conexión de ambos datos sin necesidad de copiar los datos de cada cliente manualmente.

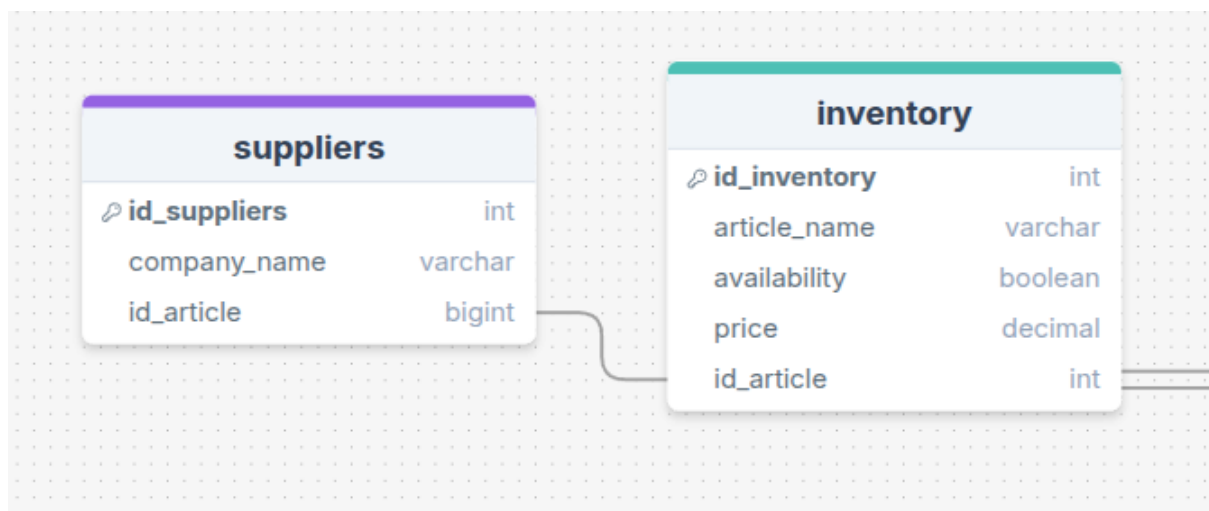
Tercera Forma Normal (3FN)

La tercera forma normal establece que si un atributo dentro de una entidad no depende directamente de su llave principal entonces se debe crear un atributo alternativo con dicho atributo y darle una llave foránea que relacione ambos atributos.

Descripción

Al crear tablas independientes que no dependen de la llave principal se da libertad de funcionamiento a la base de datos, ya que le da la posibilidad de añadirle relación a una tabla futura junto con la tabla secundaria recién creada. A veces parece que las tablas ya están estructuradas correctamente pero vale la pena revisar si la tercera forma normal se aplica del todo, porque puede que se esté omitiendo la creación de más tablas útiles en un futuro.

Gráfica



Descripción Técnica

Aquí se puede ver que ambas tablas, suppliers y inventory contienen un id_article, se podría pensar que este id_article depende de la llave primaria de cada uno pero este no es el caso. El id_article sirve para crear una tabla con todos los artículos y su descripción uno por uno, dándole una libertad importante al cliente para relacionar sus datos.

Referencias