

Base de datos relacional: Diseño



El diseño de una base de datos relacional es un proceso crucial para garantizar que los datos se almacenen de manera eficiente, sean fáciles de acceder y mantener, y se minimicen los errores y la redundancia. Para eso vamos a seguir los siguientes pasos.

1. Recolección y análisis de requisitos

Objetivo:

Comprender las necesidades y requisitos del negocio que la base de datos debe cumplir.

Ejemplo:

Supongamos que estamos diseñando una base de datos para una biblioteca en línea.

2. Identificación de las entidades y atributos (tablas y campos)

Objetivo:

Determinar las entidades principales (objetos o conceptos) sobre las cuales se almacenarán datos y los atributos (características) de cada entidad.

Ejemplo:

Para una biblioteca en línea, las entidades podrían ser: Clientes, Pedidos, Libros, Autores, Categorías.

- **Clientes:** id, Nombre, Dirección, Email, Teléfono.
- **Pedidos:** id, Fecha, ID_Cliente, Total.
- **Libros:** id, ISBN, Título, Precio, Autor_id, Categoría_id.
- **Autores:** id, Nombre.
- **Categorías:** id, Descripción.

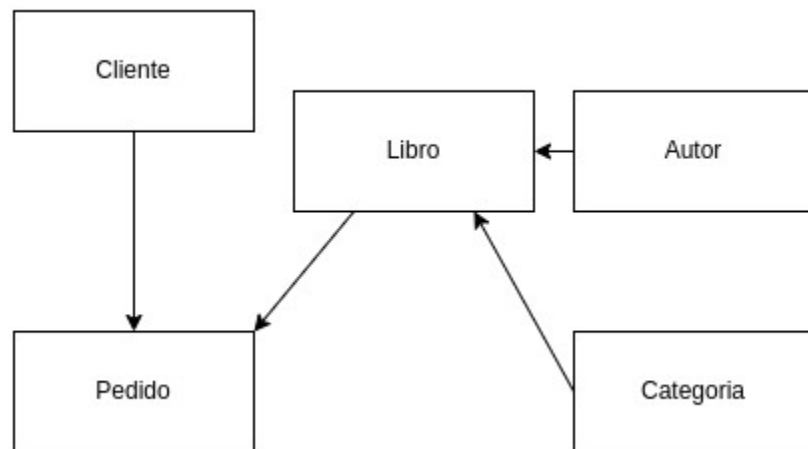
3. Creación del modelo entidad-relación (ER)

Objetivo:

Representar gráficamente las entidades y sus relaciones utilizando un diagrama ER.

Ejemplo:

- Cliente - (realiza) -> Pedido
- Pedido - (contiene) -> Libro
- Libro - (escrito por) -> Autor
- Libro - (pertenece a) -> Categoría



4. Normalización

Objetivo:

Aplicar reglas de normalización para eliminar redundancias y asegurar la integridad de los datos. Las formas normales más comunes son:

- Primera Forma Normal (1NF): Eliminar grupos repetitivos creando tablas separadas para cada conjunto de datos relacionados y uniéndolas con claves primarias.
- Segunda Forma Normal (2NF): Eliminar los datos redundantes que dependan de una parte de la clave primaria.
- Tercera Forma Normal (3NF): Eliminar los datos que no dependan directamente de la clave primaria.

Ejemplo:

- En 1NF, aseguramos que todos los atributos de una entidad son atómicos (no divisibles).
- En 2NF, nos aseguramos que todos los atributos no clave dependen de la clave primaria completa.
- En 3NF, nos aseguramos que no haya dependencias transitivas (atributos que dependen de otros atributos no clave).

5. Definición de las claves primarias y foráneas

Objetivo:

Establecer identificadores únicos para cada registro en una tabla (claves primarias) y definir cómo se relacionan las tablas entre sí (claves foráneas).

Ejemplo:

- Clave primaria para la tabla clientes: id.
- Clave foránea en la tabla pedidos: cliente_id (relacionada con Cliente).

6. Implementación del modelo en un SGBD

Objetivo:

Crear las tablas en un Sistema de Gestión de Bases de Datos (SGBD) usando el modelo relacional diseñado.

Ejemplo:

Utilizando SQL, crearemos las tablas y definimos las relaciones:

```
CREATE TABLE clientes (  
    id INT NOT NULL AUTO_INCREMENT,  
    nombre VARCHAR(100),  
    direccion VARCHAR(100),  
    email VARCHAR(100),  
    telefono VARCHAR(15),  
    PRIMARY KEY (id)  
);
```

```
CREATE TABLE autores (  
    id INT NOT NULL AUTO_INCREMENT,  
    nombre VARCHAR(100),  
    PRIMARY KEY (id)  
);
```

```
CREATE TABLE categorias (  
    id INT NOT NULL AUTO_INCREMENT,  
    descripcion VARCHAR(100),  
    PRIMARY KEY (id)  
);
```

```
CREATE TABLE pedidos (  
    id INT NOT NULL AUTO_INCREMENT,  
    fecha DATE,  
    cliente_id INT,  
    total DECIMAL(10, 2),  
    PRIMARY KEY (id)  
    FOREIGN KEY (cliente_id) REFERENCES clientes(id)  
);
```

```
CREATE TABLE libros (  
    id INT NOT NULL AUTO_INCREMENT,  
    isbn VARCHAR(20),  
    titulo VARCHAR(100),  
    precio DECIMAL(10, 2),  
    autor_id INT,  
    categoria_id INT,  
    PRIMARY KEY (id),  
    FOREIGN KEY (autor_id) REFERENCES autores(id),  
    FOREIGN KEY (categoria_id) REFERENCES categorias(id)  
);
```

7. Pruebas y ajuste de rendimiento

Objetivo:

Probar la base de datos para asegurarse de que funcione según lo esperado y ajustar el rendimiento según sea necesario.

Ejemplo:

Insertar datos de prueba en las tablas y realizar consultas para verificar la integridad referencial y el rendimiento de las consultas.

8. Documentación y mantenimiento

Objetivo:

Documentar el diseño y las estructuras de la base de datos para futuros desarrollos y mantenimiento.

Ejemplo:

Crear documentación que describa las tablas, relaciones, índices y cualquier regla de negocio aplicada.

Consideraciones adicionales

- **Seguridad:** Definir roles y permisos para proteger los datos.
- **Escalabilidad:** Asegurarse de que el diseño permita un crecimiento futuro sin grandes cambios.
- **Backup y recuperación:** Establecer procedimientos para respaldos y recuperación de datos.

Siguiendo este proceso podremos diseñar una base de datos relacional sólida y eficiente, asegurando que cumpla con las necesidades del negocio y se mantenga flexible y escalable para el futuro.