

Base de datos relacional: Tipos de relaciones



Las bases de datos relacionales son un tipo de sistema de gestión de bases de datos (SGBD) que utilizan un modelo basado en relaciones. En este modelo, los datos se organizan en tablas, y las relaciones entre estas tablas se establecen mediante claves.

Conceptos Básicos

Tablas

Una tabla es una colección de datos organizados en filas y columnas. Cada fila representa un registro único, y cada columna representa un campo dentro del registro.

Claves Primarias (Primary Keys)

Una clave primaria es un campo (o una combinación de campos) que identifica de manera única cada fila en una tabla. No puede contener valores nulos y debe ser única.

Claves Foráneas (Foreign Keys)

Una clave foránea es un campo (o una combinación de campos) en una tabla que crea un vínculo con la clave primaria de otra tabla. Esta relación permite que las tablas se vinculen entre sí.

Tipos de Relaciones

1. Relación Uno a Uno (1:1)

En una relación uno a uno, un registro en una tabla está asociado con un único registro en otra tabla, y viceversa. Esto puede utilizarse cuando una entidad se divide en dos tablas por razones de diseño.

Ejemplo: Supongamos que tenemos una tabla **personas** y otra tabla **pasaportes** donde cada persona puede tener un solo pasaporte y cada pasaporte está asociado a una sola persona.

```
CREATE TABLE personas (  
    id INT,  
    nombre VARCHAR(100),  
    apellido VARCHAR(100),  
    PRIMARY KEY (id)  
);
```

```
CREATE TABLE pasaportes (  
    id INT,  
    persona_id INT,  
    numero_pasaporte VARCHAR(50),  
    PRIMARY KEY (id)  
    FOREIGN KEY (persona_id) REFERENCES personas(id)  
);
```

2. Relación Uno a Muchos (1:n)

En una relación uno a muchos, un registro en una tabla puede estar asociado con múltiples registros en otra tabla. Este es el tipo de relación más común.

Ejemplo: Supongamos que tenemos una tabla **clientes** y una tabla **pedidos** donde cada cliente puede realizar múltiples pedidos.

```
CREATE TABLE clientes (  
    id INT,  
    nombre VARCHAR(100),  
    apellido VARCHAR(100),  
    PRIMARY KEY (id)  
);  
  
CREATE TABLE pedidos (  
    id INT,  
    cliente_id INT,  
    fecha_pedido DATE,  
    PRIMARY KEY (id),  
    FOREIGN KEY (cliente_id) REFERENCES clientes(id)  
);
```


3. Relación Muchos a Muchos (n:n)

En una relación muchos a muchos, múltiples registros en una tabla pueden estar asociados con múltiples registros en otra tabla. Para implementar esta relación, generalmente se utiliza una tabla intermedia que contiene las claves foráneas de ambas tablas.

Ejemplo: Supongamos que tenemos una tabla `estudiantes` y una tabla `cursos` donde cada estudiante puede inscribirse en múltiples cursos y cada curso puede tener múltiples estudiantes.

La tabla `inscripciones` sería la tabla intermedia, la combinación de `estudiante_id` y `curso_id` actúa como clave primaria compuesta, y estas columnas también actúan como claves foráneas hacia las tablas `estudiantes` y `cursos`, respectivamente.

```
CREATE TABLE estudiantes (  
    id INT,  
    nombre VARCHAR(100),  
    apellido VARCHAR(100),  
    PRIMARY KEY (id)  
);  
  
CREATE TABLE cursos (  
    id INT,  
    nombre_curso VARCHAR(100),  
    PRIMARY KEY (id)  
);  
  
CREATE TABLE inscripciones (  
    estudiante_id INT,  
    curso_id INT,  
    fecha_inscripcion DATE,  
    FOREIGN KEY (estudiante_id) REFERENCES estudiantes(id),  
    FOREIGN KEY (curso_id) REFERENCES cursos(id)  
);
```

Conclusión

Las relaciones en bases de datos relacionales permiten organizar y conectar datos de manera eficiente, asegurando integridad y consistencia. Comprender cómo funcionan estas relaciones y cómo implementarlas correctamente es esencial para diseñar bases de datos robustas y funcionales.