

Bases de Datos Relacionales

Sesión 3: UPEMOR

Objetivos de Aprendizaje

- **Saber (Dimensión Conceptual):**
Describir las características y elementos del modelo de bases de datos relacional.
- **Saber Hacer (Dimensión Actuacional):**
Modelar bases de datos relacionales utilizando entidades, atributos, relaciones y llaves.

Introducción

- Las bases de datos relacionales son uno de los modelos más usados en el desarrollo de sistemas de información. Su diseño permite estructurar la información en tablas relacionadas entre sí, lo que facilita la organización, consulta y actualización de los datos de manera eficiente.

¿Qué es una base de datos relacional?

- Una **base de datos relacional** es un conjunto de datos estructurados en **tablas (relaciones)** que se conectan mediante **llaves**. Fue propuesta por **Edgar F. Codd** en 1970 como un modelo lógico basado en teoría de conjuntos.
 - Cada tabla representa una **entidad** o concepto del mundo real.
 - Cada fila (registro) representa una **instancia** de esa entidad.
 - Cada columna representa un **atributo**.

Elementos principales del modelo relacional

Elemento	Descripción
Tabla (relación)	Conjunto de datos estructurados en filas y columnas.
Atributo (columna)	Característica o propiedad de la entidad.
Tupla (fila)	Registro único de datos.
Llave Primaria	Identificador único para cada registro.
Llave foránea	Atributo que crea una relación entre dos tablas.
Dominio	Conjunto de valores posibles para un atributo.
Esquema	Estructura de toda la base de datos.

Características del modelo relacional



Ventajas de las bases de datos relacionales

- ❑ Estructura clara y normalizada
- ❑ Facilitan la consistencia y precisión de los datos
- ❑ Lenguaje estándar (SQL) para su manipulación
- ❑ Soportadas por múltiples sistemas (MySQL, PostgreSQL, SQL Server, Oracle)
- ❑ Permiten el diseño modular de sistemas de información

Diseño lógico relacional: paso a paso

Identificar entidades del mundo real (por ejemplo, Cliente, Producto, Pedido).

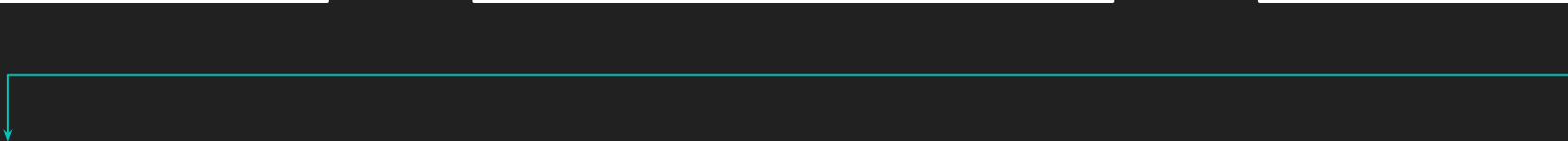
Definir atributos de cada entidad (nombre, fecha, cantidad, etc.).

Establecer relaciones entre entidades (uno a uno, uno a muchos, muchos a muchos).

Determinar llaves primarias y llaves foráneas.

Normalizar las tablas para evitar redundancias.

Representar en esquema relacional (tablas conectadas por llaves).



Ejemplo práctico: Tienda en línea

■ Entidades:

- Cliente (id_cliente, nombre, email)
- Producto (id_producto, nombre, precio)
- Pedido (id_pedido, fecha, id_cliente)
- DetallePedido (id_pedido, id_producto, cantidad)

■ Relaciones:

- Un cliente puede hacer varios pedidos → relación 1:N
- Un pedido puede incluir varios productos y un producto puede estar en varios pedidos → relación N:M (con tabla intermedia)

¿Por qué necesitamos una **tabla intermedia** en la relación **N:M**?

□ Relación N:M:

- Cuando tenemos una **relación N:M**, significa que un registro de una tabla puede estar relacionado con muchos registros de otra tabla, y viceversa.

□ Por ejemplo:

- **Pedido** → Un pedido puede tener varios productos.
- **Producto** → Un producto puede aparecer en muchos pedidos.

¿Por qué necesitamos una **tabla intermedia** en la relación **N:M**?

□ Problema:

- Si intentamos almacenar esta relación directamente en las tablas de **Pedido** o **Producto**, la base de datos no sabría cómo gestionar varias asociaciones entre estas tablas, porque una fila de **Pedido** podría tener múltiples productos, y un producto podría estar en varios pedidos al mismo tiempo.

□ Solución: Tabla Intermedia

- La **tabla intermedia** se utiliza para "**dividir**" la relación **N:M** en dos relaciones **1:N**.

¿Por qué necesitamos una **tabla intermedia** en la relación **N:M**?

□ Ejemplo:

- La tabla **Pedido** tiene un **id_pedido** único.
- La tabla **Producto** tiene un **id_producto** único.
- La tabla **DetallePedido** (intermedia) asocia cada **id_pedido** con los productos correspondientes y la **cantidad** de cada uno.

□ Esquema:

- Un **Pedido** puede estar asociado a muchos **Productos**.
- Un **Producto** puede estar asociado a muchos **Pedidos**.
- La **tabla intermedia** (DetallePedido) resuelve esto al almacenar qué productos están en qué pedidos y la cantidad de cada producto.

□ Beneficio:

- La tabla intermedia ayuda a **mantener la estructura organizada y facilita las consultas** sin duplicar datos, y asegura la integridad referencial.

Conclusiones clave

- ❑ Las bases de datos relacionales son el estándar para almacenar información estructurada.
- ❑ Su fortaleza está en la organización mediante tablas y en el uso de llaves para conectar información.
- ❑ El diseño adecuado es esencial para mantener la integridad, escalabilidad y eficiencia del sistema.