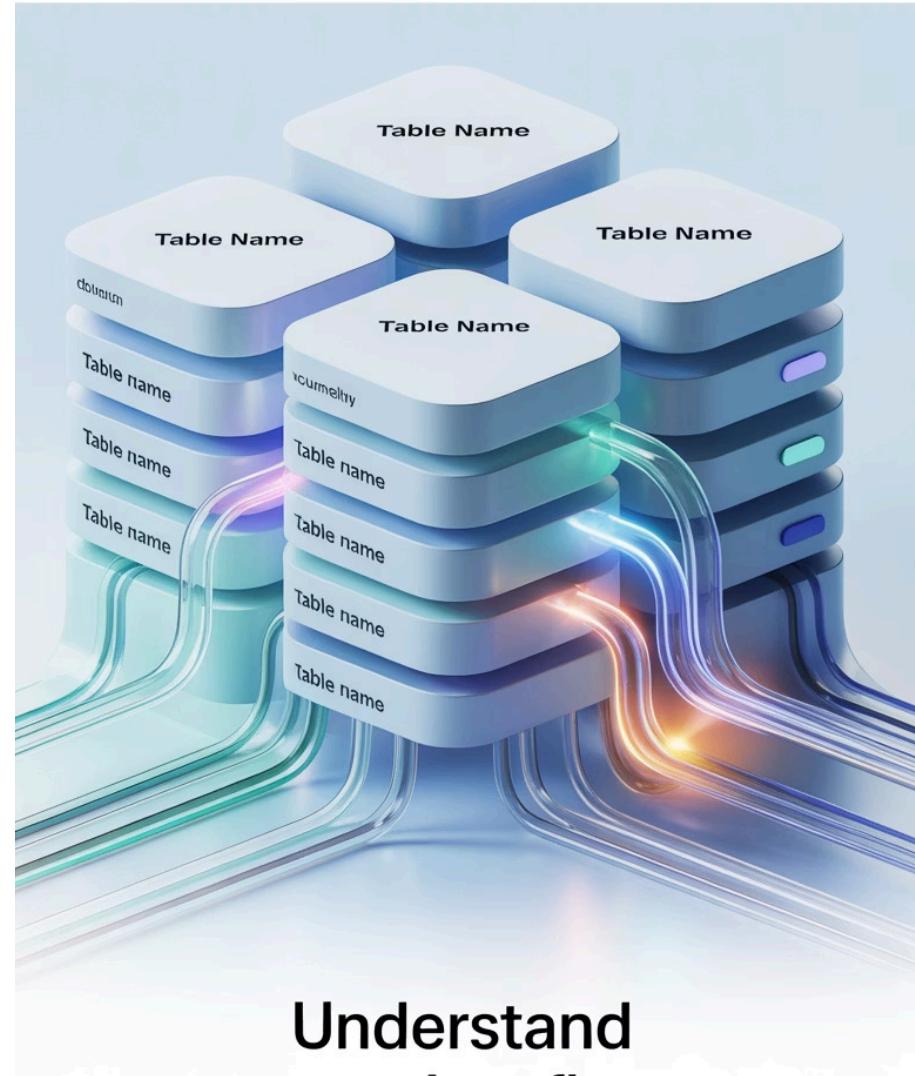


# El modelo relacional: Estructura, claves e integridad de los datos

Bienvenido a la Unidad 3 de nuestro curso de bases de datos. En este módulo, exploraremos la base teórica de la mayoría de los sistemas de bases de datos modernos como MySQL y PostgreSQL: el modelo relacional. Examinaremos su estructura, comprenderemos la importancia de las claves y aprenderemos cómo se mantiene la integridad de los datos en las bases de datos relacionales.



**Understand  
your data flow,**

Explore data →

Download Schema →

# Agenda

1

## Introducción al modelo relacional

Origen, conceptos básicos e importancia en los sistemas de bases de datos modernos

2

## Componentes y estructura

Tablas, filas, columnas y su terminología formal

3

## Claves y relaciones

Claves primarias, claves externas y relaciones entre tablas

4

## Reglas de integridad

Integridad de dominio, integridad referencial y aplicaciones prácticas

5

## Aplicaciones y actividades

Ejercicios prácticos y próximos pasos en el aprendizaje de bases de datos

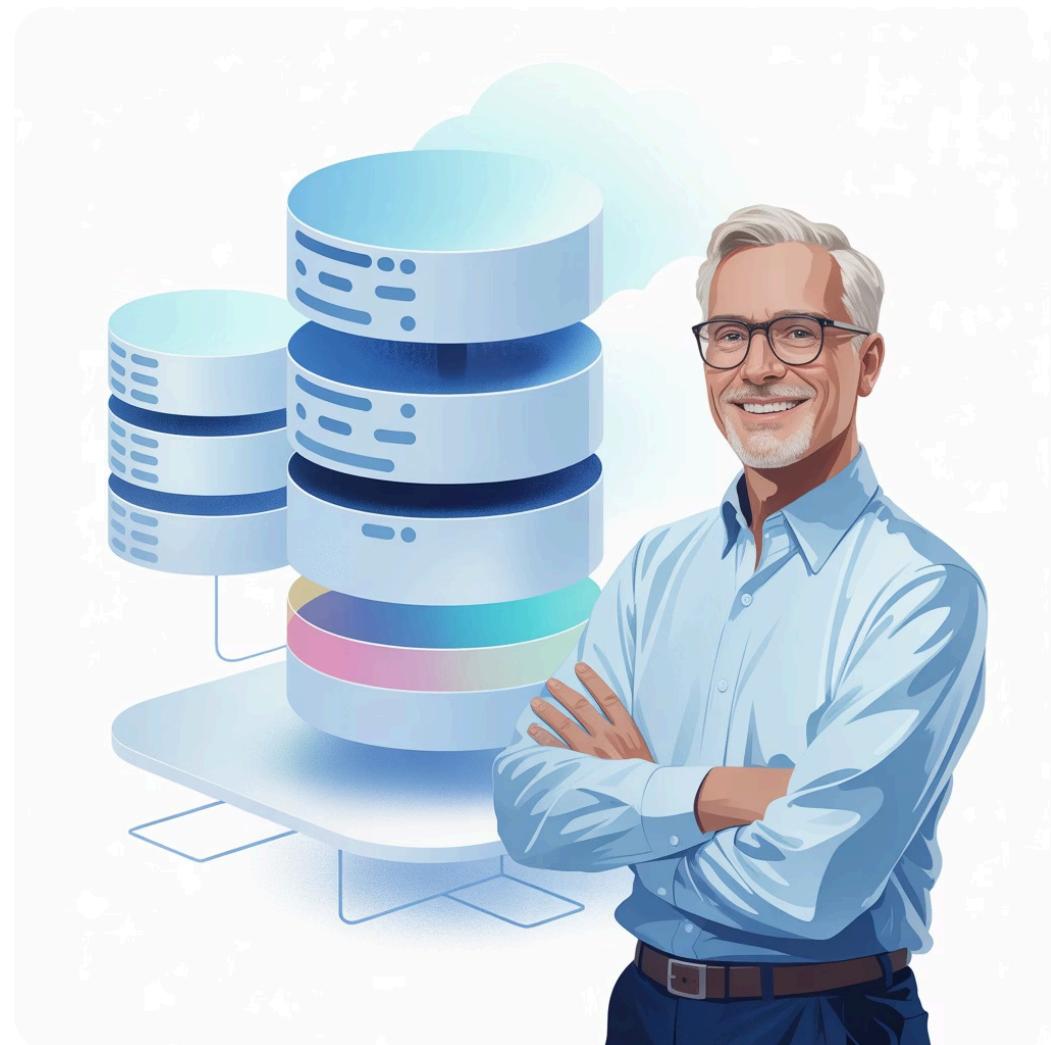
# ¿Qué es el modelo relacional?

## Contexto histórico

Propuesto por Edgar F. Codd en 1970 mientras trabajaba en IBM, el modelo relacional revolucionó la gestión de datos al proporcionar una base matemática para los sistemas de bases de datos.

## Innovación clave

Antes del modelo relacional, los sistemas de bases de datos utilizaban modelos jerárquicos o de red, que eran menos flexibles y más complejos de mantener.



El modelo relacional organiza los datos en tablas (relaciones) donde cada fila representa un registro y cada columna representa un atributo de ese registro.

# Componentes centrales del modelo relacional

## Tabla (Relación)

Una colección de datos relacionados organizados en filas y columnas. Cada tabla normalmente representa una entidad o una relación entre entidades.

Ejemplo: Estudiantes, Cursos, Inscripciones

## Fila (Tupla)

Un único registro dentro de una tabla que contiene un conjunto de elementos de datos relacionados. Cada fila representa una instancia de la entidad.

Ejemplo: La información de un estudiante específico

## Columna (Atributo)

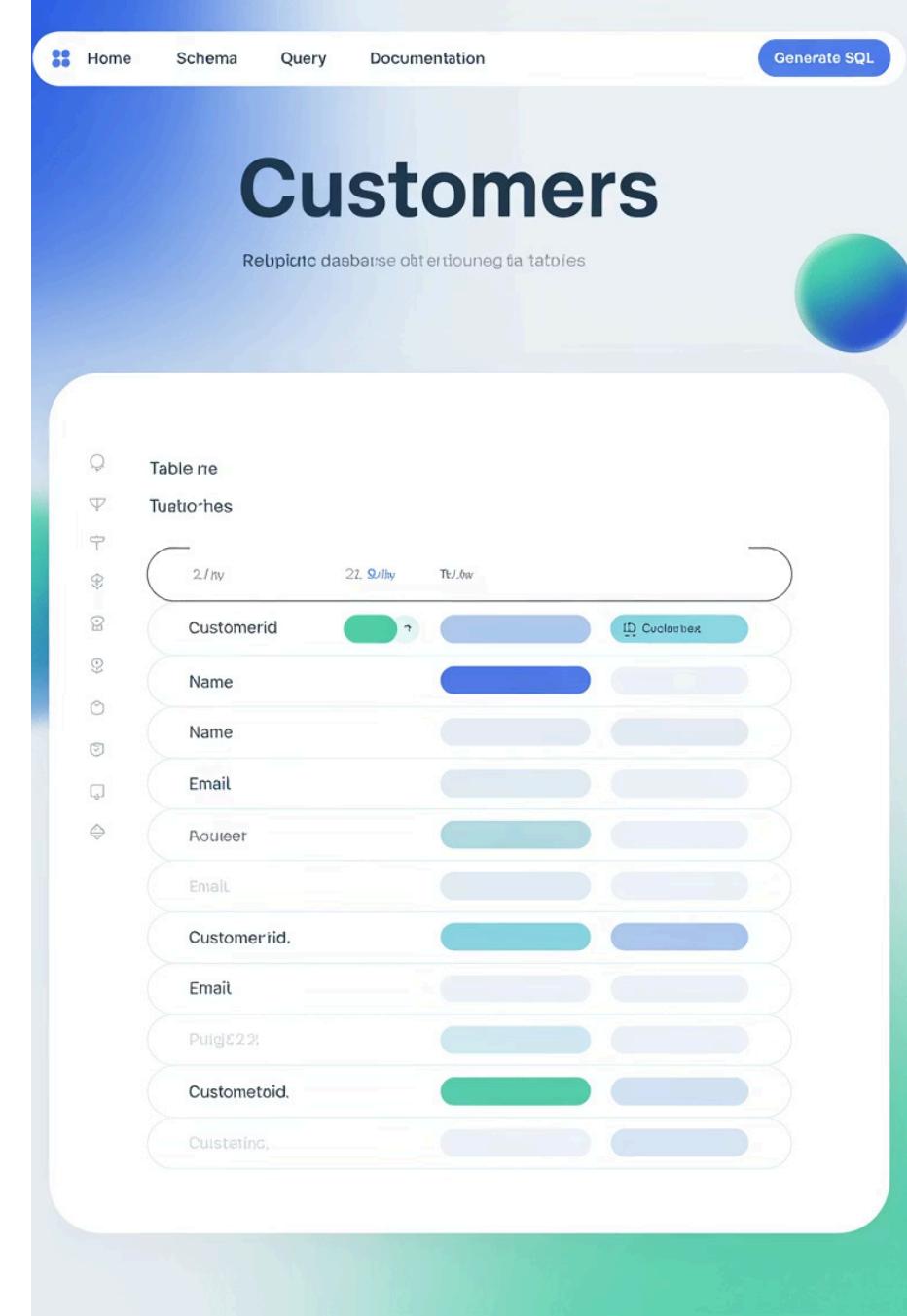
Una característica o propiedad específica de la entidad representada por la tabla. Cada columna tiene un tipo de datos definido.

Ejemplo: ID del estudiante, Nombre, Correo electrónico

Comprender estos componentes y su terminología formal es esencial para un diseño de base de datos eficaz y la comunicación entre los profesionales de bases de datos.

# Representación visual de los componentes relacionales

En el modelo relacional, utilizamos tanto términos informales (tablas, filas, columnas) como términos formales (relaciones, tuplas, atributos). Ambas terminologías se utilizan ampliamente en la comunidad de bases de datos, y comprender ambos conjuntos ayuda al leer documentación técnica o artículos académicos.



# Tipos de claves en bases de datos relacionales

## Clave primaria

Una columna o combinación de columnas que identifica de forma única cada fila de una tabla. Cada tabla debe tener una clave primaria.

- Debe ser única en todas las filas
- No puede contener valores NULL
- Rara vez o nunca debe cambiar

Ejemplo: StudentID en una tabla de estudiantes

## Clave candidata

Cualquier columna o conjunto de columnas que podría servir potencialmente como clave primaria. Una tabla puede tener varias claves candidatas.

- Debe ser única y no NULL
- La clave primaria se selecciona de entre las claves candidatas

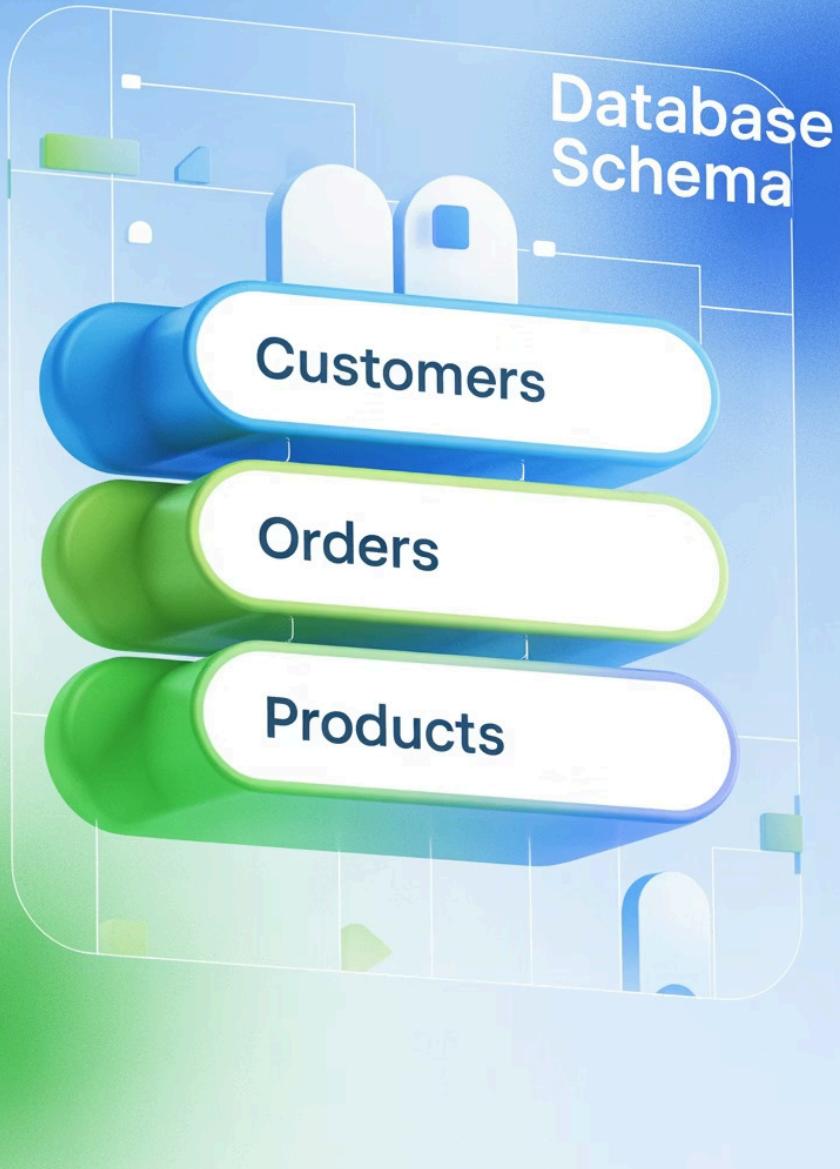
Ejemplo: Dirección de correo electrónico (si es única para cada estudiante)

## Clave externa

Una columna o conjunto de columnas que hace referencia a la clave primaria de otra tabla, estableciendo una relación entre las tablas.

- Crea enlaces entre tablas
- Aplica la integridad referencial

Ejemplo: DepartmentID en una tabla de estudiantes que hace referencia a la tabla Departments



## Relaciones entre tablas

Las relaciones están en el corazón del modelo relacional, lo que nos permite conectar datos relacionados entre tablas. Estas conexiones se establecen mediante claves externas que hacen referencia a las claves primarias de otras tablas.

- **Uno a uno:** Cada fila de la Tabla A se relaciona con exactamente una fila de la Tabla B
- **Uno a muchos:** Cada fila de la Tabla A se relaciona con varias filas de la Tabla B
- **Muchos a muchos:** Varias filas de la Tabla A se relacionan con varias filas de la Tabla B (requiere una tabla de unión)

# Ejemplo de relaciones de tablas

Tabla de estudiantes

ID de estudiante (PK)	Nombre	ID de especialidad (FK)
1001	Ana García	CS100
1002	Carlos López	ENG200
1003	María Rodríguez	CS100

Tabla de especialidades

ID de especialidad (PK)	Nombre	Departamento
CS100	Ciencias de la computación	Ingeniería
ENG200	Ingeniería civil	Ingeniería
BUS300	Administración de empresas	Negocios

En este ejemplo, la tabla Estudiantes tiene una clave externa (ID de especialidad) que hace referencia a la clave principal de la tabla Especialidades, lo que crea una relación de uno a muchos.

# Reglas de integridad en el modelo relacional

1

## Integridad del dominio

Garantiza que todos los valores de una columna se ajusten al tipo de datos y las restricciones definidos.

- Validación del tipo de datos (enteros, cadenas, fechas)
- Comprobaciones de rango (por ejemplo, edad entre 18 y 65 años)
- Validación de formato (por ejemplo, formato de dirección de correo electrónico)

Ejemplo: Evitar la entrada de texto en un campo GPA numérico

2

## Integridad de la entidad

Garantiza que cada fila de una tabla se identifique de forma única.

- La clave principal debe ser única
- La clave principal no puede ser NULL

Ejemplo: Cada estudiante debe tener un ID único

3

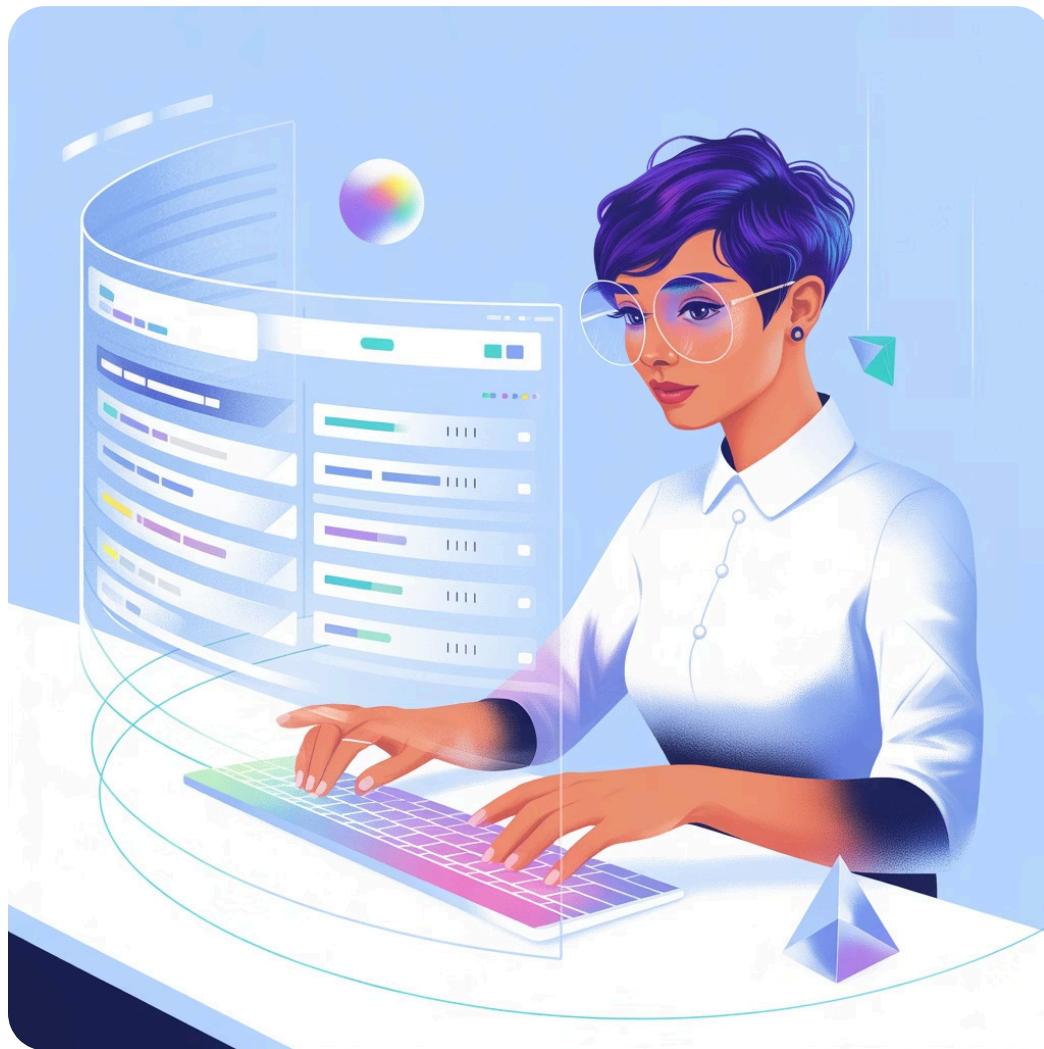
## Integridad referencial

Garantiza que las relaciones entre las tablas sigan siendo coherentes.

- Los valores de la clave externa deben coincidir con una clave principal existente
- Evita registros huérfanos

Ejemplo: No se puede asignar un estudiante a una especialidad inexistente

# Integridad del dominio en la práctica



## Ejemplos de implementación

- Restricciones de tipo de datos (INT, VARCHAR, DATE)
- Restricciones NOT NULL
- Restricciones CHECK (p. ej., Calificación ENTRE 0 Y 100)
- Valores DEFAULT
- Restricciones UNIQUE

La integridad del dominio garantiza la calidad de los datos al evitar que se almacenen valores no válidos en la base de datos. Esta es la primera línea de defensa contra la corrupción de datos.

## "Cascading data flow"



# Integridad referencial en la práctica

## Manejo de cambios en datos relacionados

- 1
- 2
- 3
- 4

### CASCADE

Los cambios en la tabla principal se propagan automáticamente a la tabla secundaria.

### RESTRICT o NO ACTION

Evita cambios en los registros principales si existen registros secundarios.

### SET NULL

Establece la clave externa en NULL cuando se elimina la fila a la que se hace referencia.

### SET DEFAULT

Establece la clave externa en un valor predeterminado cuando se elimina la fila a la que se hace referencia.

# Ejemplo práctico de integridad referencial

## Escenario: Eliminar una especialidad

¿Qué sucede si intentamos eliminar Ciencias de la Computación (CS100) de la tabla de especialidades?

- **Con RESTRICT:** La eliminación se bloquearía porque hay estudiantes asignados a ella.
- **Con CASCADE:** CS100 y todos los estudiantes con esa especialidad se eliminarían.
- **Con SET NULL:** CS100 se eliminaría y las especialidades de los estudiantes se establecerían en NULL.



La integridad referencial ayuda a prevenir inconsistencias en los datos al garantizar que las relaciones entre las tablas sigan siendo válidas durante las operaciones de la base de datos.

# Ventajas del modelo relacional

## Simplicidad lógica

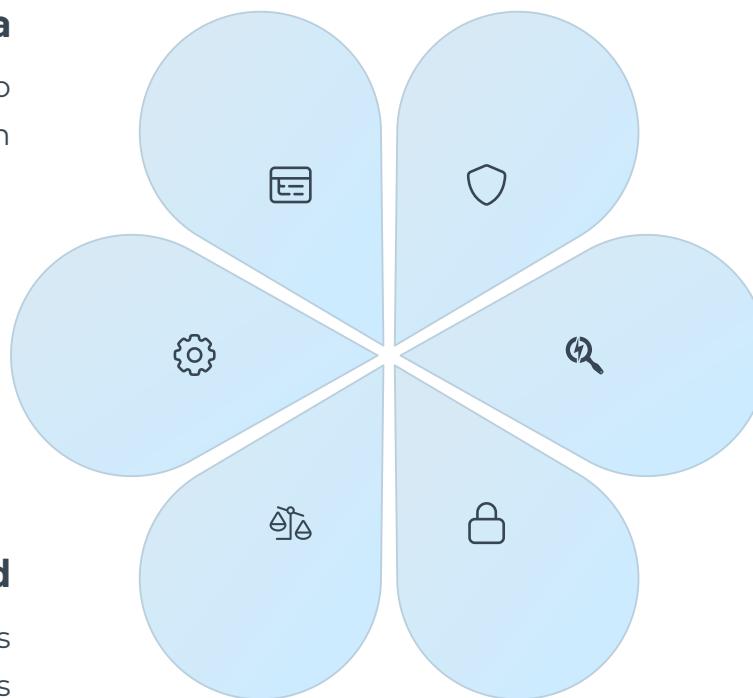
Estructura tabular intuitiva que refleja cómo organizamos naturalmente la información

## Soporte generalizado

Amplia adopción por parte de los proveedores y herramientas modernos de DBMS

## Escalabilidad

Capacidad para manejar cantidades crecientes de datos y usuarios



## Integridad de datos

Mecanismos integrados para mantener la coherencia y la precisión de los datos

## Flexibilidad de consulta

Potentes capacidades de consulta a través de SQL y álgebra relacional

## Seguridad de datos

Control de acceso granular y gestión de permisos

# Desafíos del modelo relacional

## Consideraciones de rendimiento

- Las operaciones JOIN pueden resultar costosas con grandes conjuntos de datos
- La normalización puede introducir complejidad en las consultas
- No es óptimo para datos jerárquicos o tipo red

Estas limitaciones han llevado al desarrollo de bases de datos NoSQL y NewSQL para casos de uso específicos, aunque las bases de datos relacionales siguen siendo dominantes para los sistemas transaccionales.



Comprender estas limitaciones ayuda a los diseñadores de bases de datos a tomar decisiones informadas sobre cuándo utilizar bases de datos relacionales y cuándo modelos alternativos podrían ser más apropiados.

# Aplicaciones en el mundo real



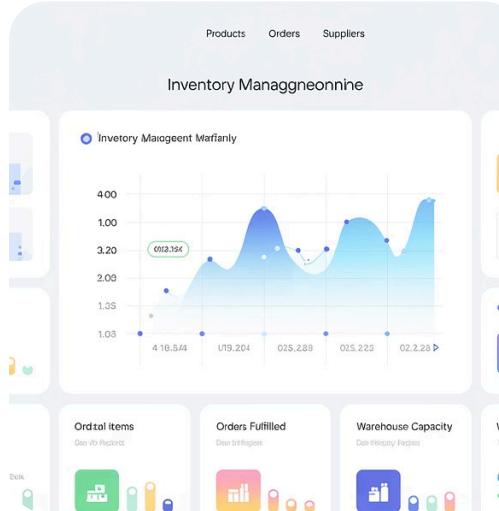
## Sistemas bancarios

Seguimiento de cuentas, transacciones y relaciones con los clientes donde la integridad de los datos es fundamental



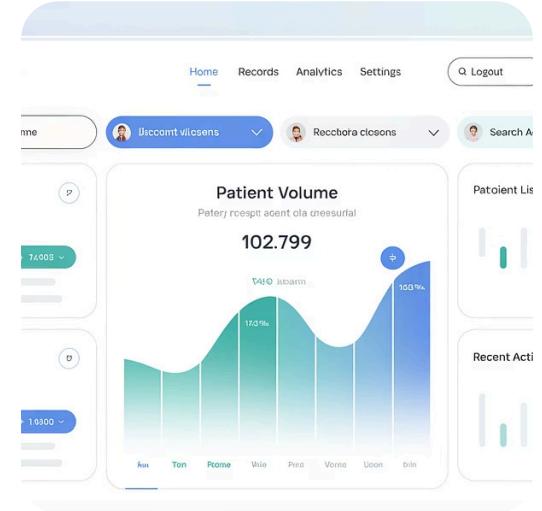
## Instituciones educativas

Gestión de registros de estudiantes, inscripciones a cursos y rendimiento académico



## Inventario minorista

Mantenimiento de catálogos de productos, niveles de inventario y transacciones de ventas



## Registros de atención médica

Almacenamiento de información del paciente, historiales médicos y planes de tratamiento

# Sistemas de gestión de bases de datos relacionales populares

## MySQL

RDBMS de código abierto conocido por su velocidad, confiabilidad y facilidad de uso. Ampliamente utilizado en aplicaciones web.

- Impulsa WordPress, Facebook, Twitter
- Excelente para cargas de trabajo con muchas lecturas

## PostgreSQL

RDBMS avanzado de código abierto con un sólido cumplimiento de los estándares y amplias funciones.

- Soporte para JSON, datos geoespaciales
- Capacidades avanzadas de indexación

## Oracle Database

RDBMS de clase empresarial con funciones integrales para grandes organizaciones.

- Alta disponibilidad y escalabilidad
- Funciones de seguridad avanzadas

## Microsoft SQL Server

RDBMS comercial con una estrecha integración con el ecosistema de Microsoft.

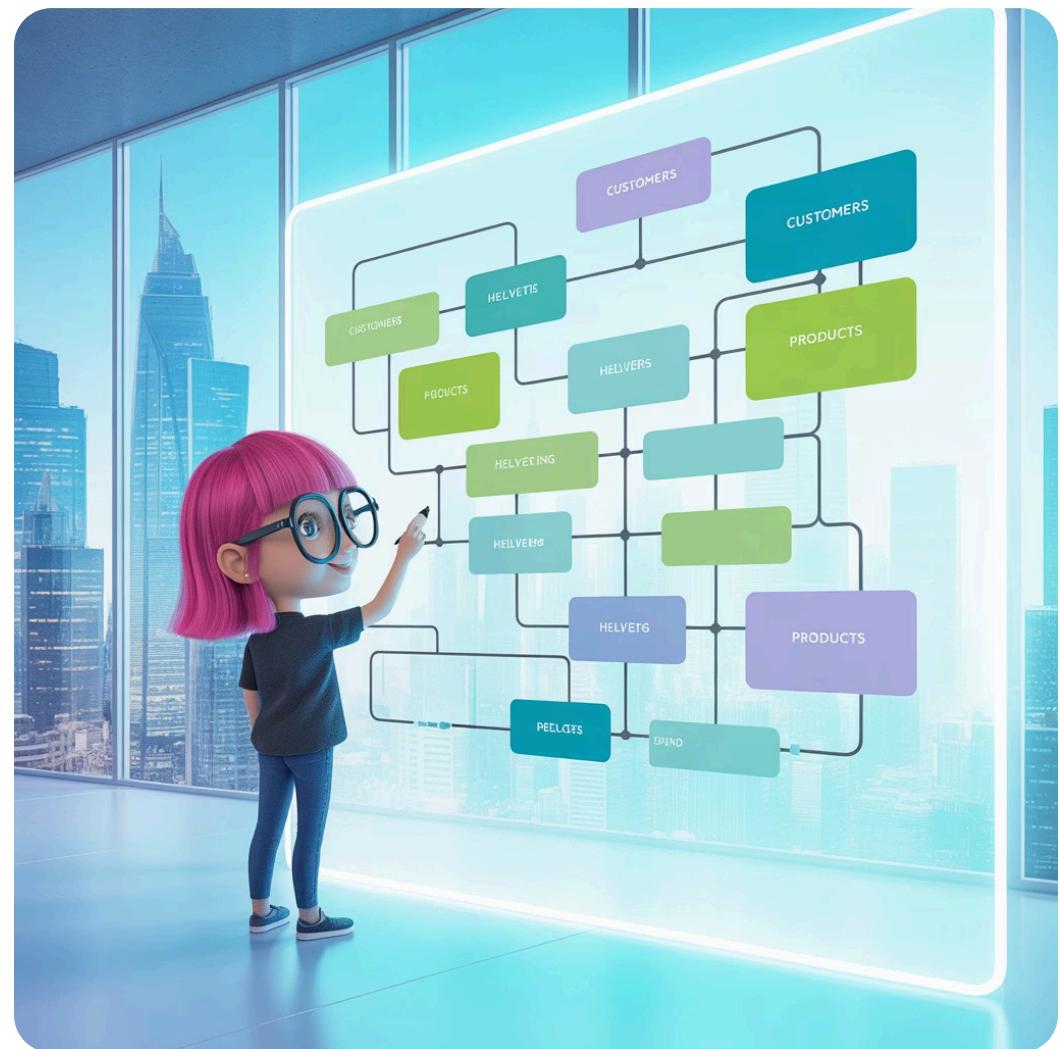
- Capacidades de inteligencia empresarial
- Excelente para entornos Windows

Cada sistema implementa el modelo relacional con diferentes características, optimizaciones y compensaciones.

# Diseñando tu primera base de datos relacional

## Enfoque paso a paso

1. Identifica las entidades (cosas a rastrear)
2. Determina los atributos para cada entidad
3. Establece relaciones entre entidades
4. Asigna claves primarias a cada tabla
5. Implementa claves externas para las relaciones
6. Aplica reglas de normalización
7. Agrega restricciones para la integridad



Un buen diseño de base de datos requiere una planificación cuidadosa y la comprensión de la estructura y las relaciones de los datos. Comienza poco a poco con un alcance claro y refina a medida que aprendes.

# Actividad práctica: Crear un modelo relacional simple

## Tarea

Diseñe un modelo relacional con al menos dos tablas relacionadas:

### Tabla de clientes

- ID de cliente (clave principal)
- Nombre
- Ciudad
- Correo electrónico

### Tabla de pedidos

- ID de pedido (clave principal)
- Fecha
- Total
- ID de cliente (clave externa)

Defina las claves primarias y externas apropiadas, y considere qué sucede cuando se elimina un cliente. ¿Cómo manejaría la integridad referencial en este caso?



# Conclusiones clave



## Estructura fundamental

El modelo relacional organiza los datos en tablas (relaciones) con filas (tuplas) y columnas (atributos), proporcionando una forma lógica e intuitiva de representar los datos.



## Reglas de integridad

Las restricciones de dominio, entidad e integridad referencial trabajan juntas para mantener la consistencia y precisión de los datos, previniendo errores e inconsistencias en la base de datos.



## Importancia clave

Las claves primarias aseguran la identificación única de los registros, mientras que las claves externas establecen y mantienen las relaciones entre las tablas, formando la columna vertebral de las bases de datos relacionales.



## Relevancia práctica

A pesar del surgimiento de modelos de bases de datos alternativos, el modelo relacional sigue siendo dominante debido a su simplicidad lógica, sus potentes capacidades de consulta y sus sólidas características de integridad de datos.