

Modelado Conceptual de Bases de Datos y Diagramas Entidad-Relación

Bienvenido a la Unidad 2 de Base de Datos I. En esta presentación, exploraremos el modelo Entidad-Relación, una herramienta fundamental para el diseño conceptual de bases de datos que nos ayuda a representar las relaciones de datos del mundo real antes de la implementación.



Agenda

1

Modelos conceptuales

Definición y propósito del modelado conceptual de datos

2

Modelo Entidad-Relación

Componentes centrales y el enfoque de Peter Chen

3

Diagramas ER

Creación e interpretación de diagramas de entidad-relación

4

Restricciones de integridad

Claves primarias, claves externas e integridad referencial

5

Aplicación práctica

Diseño de un diagrama ER para un sistema de biblioteca

¿Qué es un modelo conceptual?

Un modelo conceptual es una representación abstracta de la realidad que nos permite comprender sistemas complejos antes de la implementación. En el diseño de bases de datos, sirve como el plano que guía todas las fases de desarrollo posteriores.

Abstracción de la realidad

Captura los elementos esenciales del mundo real e ignora los detalles innecesarios

Herramienta de planificación

Permite una planificación exhaustiva antes de construir la base de datos real

Medio de comunicación

Facilita una comunicación clara entre las partes interesadas y los desarrolladores



El modelo Entidad-Relación

Contexto histórico

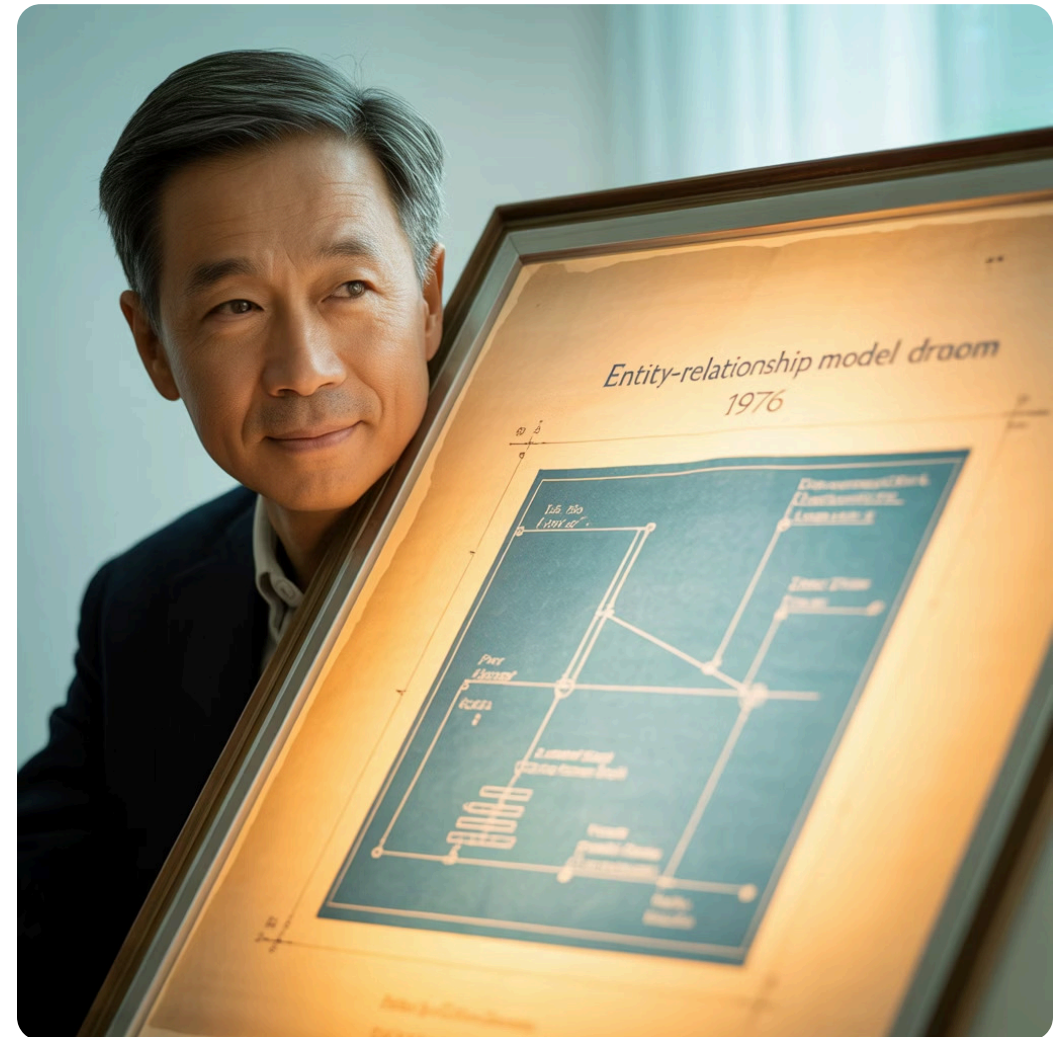
Propuesto por Peter Chen en 1976, el modelo E-R se ha convertido en la base de las metodologías modernas de diseño de bases de datos.

Propósito

Proporciona un enfoque estandarizado para representar estructuras de datos y relaciones utilizando un lenguaje visual que une la brecha entre los requisitos comerciales y la implementación técnica.

Base para el DER

El modelo E-R proporciona la base teórica para crear diagramas de entidad-relación (DER), la representación visual de las estructuras de la base de datos.



El modelo E-R de Peter Chen revolucionó la forma en que conceptualizamos y diseñamos bases de datos al proporcionar un marco visual para representar relaciones de datos complejas.

Conceptos básicos del modelo E-R



Entidades

Objetos o conceptos del mundo real que pueden identificarse de forma inequívoca.

Ejemplo: Estudiante, Profesor, Curso

Representado como rectángulos en los diagramas ER.



Atributos

Propiedades o características que describen las entidades.

Ejemplo: Estudiante (ID, Nombre, Edad, Correo electrónico)

Representado como óvalos o elipses en los diagramas ER.



Relaciones

Asociaciones entre dos o más entidades.

Ejemplo: Estudiante "se inscribe en" Curso

Representado como diamantes en los diagramas ER.

Estos tres elementos centrales forman la base de cualquier modelo E-R y son esenciales para representar con precisión las estructuras de datos del mundo real.

Tipos de atributos

- **Atributos simples**

No se pueden dividir más (Ej: ID del estudiante)

- **Atributos compuestos**

Se pueden dividir en partes más pequeñas (Ej: Dirección: calle, ciudad, código postal)

- **Atributos de un solo valor**

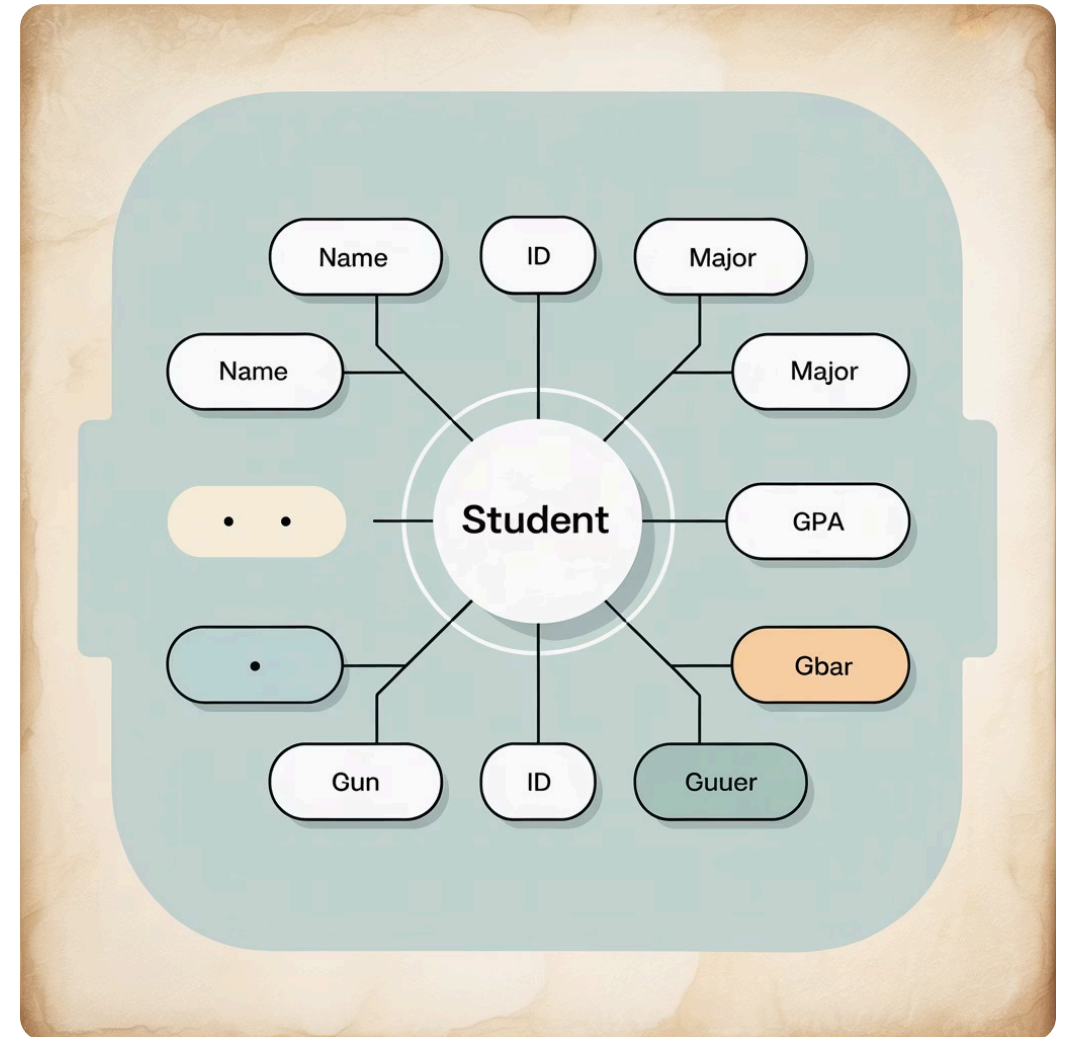
Contienen solo un valor (Ej: Fecha de nacimiento)

- **Atributos de valores múltiples**

Pueden contener varios valores (Ej: Números de teléfono)

- **Atributos derivados**

Valores calculados a partir de otros atributos (Ej: Edad a partir de la fecha de nacimiento)



Comprender los tipos de atributos es crucial para crear modelos de bases de datos precisos y eficientes que representen correctamente las características de los datos del mundo real.

Tipos de relaciones



Uno a uno (1:1)

En una relación uno a uno, cada entidad de un conjunto se asocia con exactamente una entidad del otro conjunto, y viceversa.

Ninguna entidad puede existir sin su correspondiente pareja en el conjunto opuesto.

Ejemplo: Persona — tiene — Identificación nacional

Representado con "1" en ambos extremos de la línea de relación.



Uno a muchos (1:N)

Cada entidad en el primer conjunto puede estar relacionada con muchas entidades en el segundo conjunto, pero cada entidad en el segundo conjunto está relacionada con solo una entidad en el primer conjunto.

Ejemplo: Departamento — emplea — Empleado

Representado con "1" en un extremo y "N" o una pata de gallo en el otro.



Muchos a muchos (N:M)

Cada entidad en el primer conjunto puede estar relacionada con muchas entidades en el segundo conjunto, y viceversa.

Ejemplo: Estudiante — se inscribe en — Curso

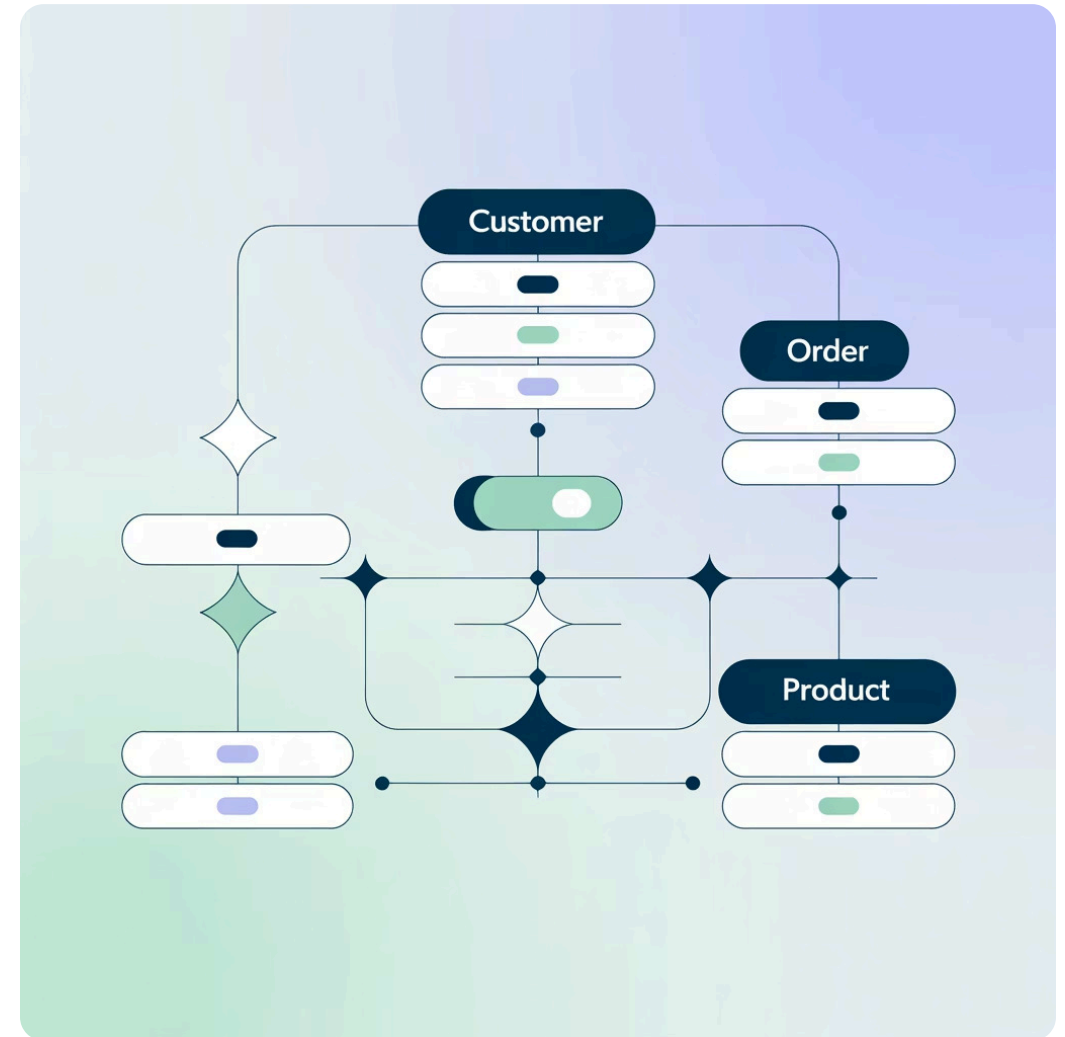
Representado con "N" o patas de gallo en ambos extremos de la relación.

Creación de diagramas de entidad-relación (DER)

Elementos de notación estándar

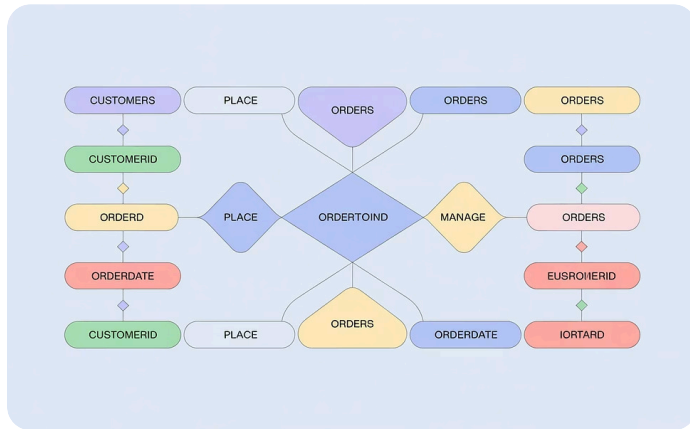
- Los rectángulos representan entidades
- Las elipses u óvalos representan atributos
- Los diamantes representan relaciones
- Las líneas conectan entidades con relaciones y atributos
- Las notaciones de cardinalidad muestran los tipos de relación (1:1, 1:N, N:M)

Los DER proporcionan un lenguaje visual estandarizado que permite a los diseñadores de bases de datos comunicar estructuras de datos complejas de forma clara e inequívoca.



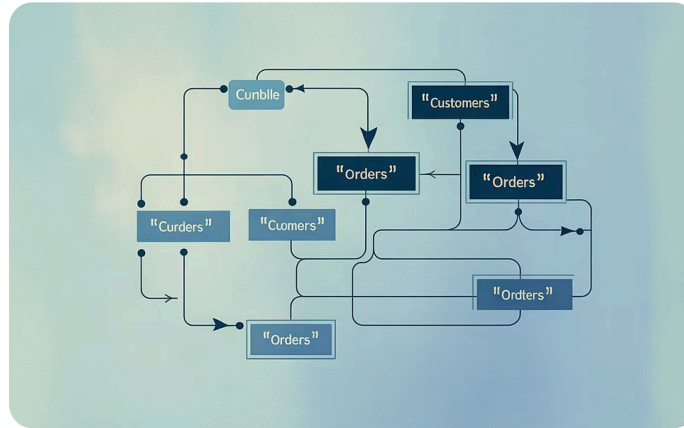
La naturaleza visual de los DER los convierte en herramientas poderosas para que tanto las partes interesadas técnicas como las no técnicas comprendan la estructura de la base de datos.

Estilos de notación DER



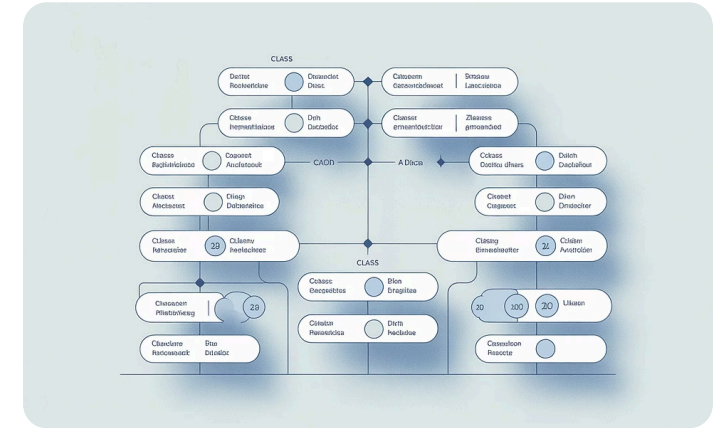
Notación Chen

La notación original creada por Peter Chen. Utiliza diamantes para las relaciones y óvalos para los atributos. Considerada la más completa, pero a veces visualmente compleja.



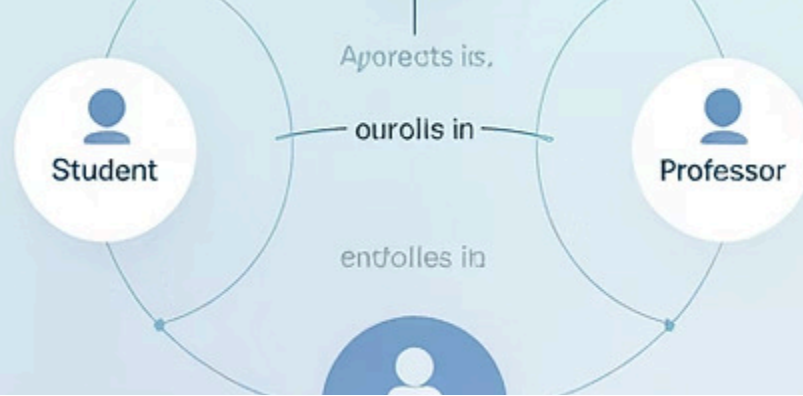
Notación de pata de gallo

Utiliza símbolos distintivos de "pata de gallo" para indicar muchas relaciones. Popular en la industria por su claridad visual y su representación intuitiva de la cardinalidad.



Diagramas de clases UML

Utiliza la notación del Lenguaje de Modelado Unificado. Representa las entidades como clases con atributos y operaciones. Comúnmente utilizado en el diseño de bases de datos orientadas a objetos.



Ejemplo práctico: Sistema académico

Entidades

- Estudiante (ID, Nombre, Correo electrónico, Especialidad)
- Curso (Código, Título, Créditos, Descripción)
- Profesor (ID, Nombre, Departamento, Oficina)

Relaciones

- Estudiante "se inscribe en" Curso (N:M)
- Profesor "enseña" Curso (1:N)
- Estudiante "asesorado por" Profesor (N:1)

Este diagrama ERD simplificado del sistema académico demuestra cómo interactúan las entidades en un escenario del mundo real. Observe cómo la relación de muchos a muchos entre Estudiante y Curso normalmente requeriría una entidad de intersección (Inscripción) en el modelo lógico.

Claves primarias

Definición

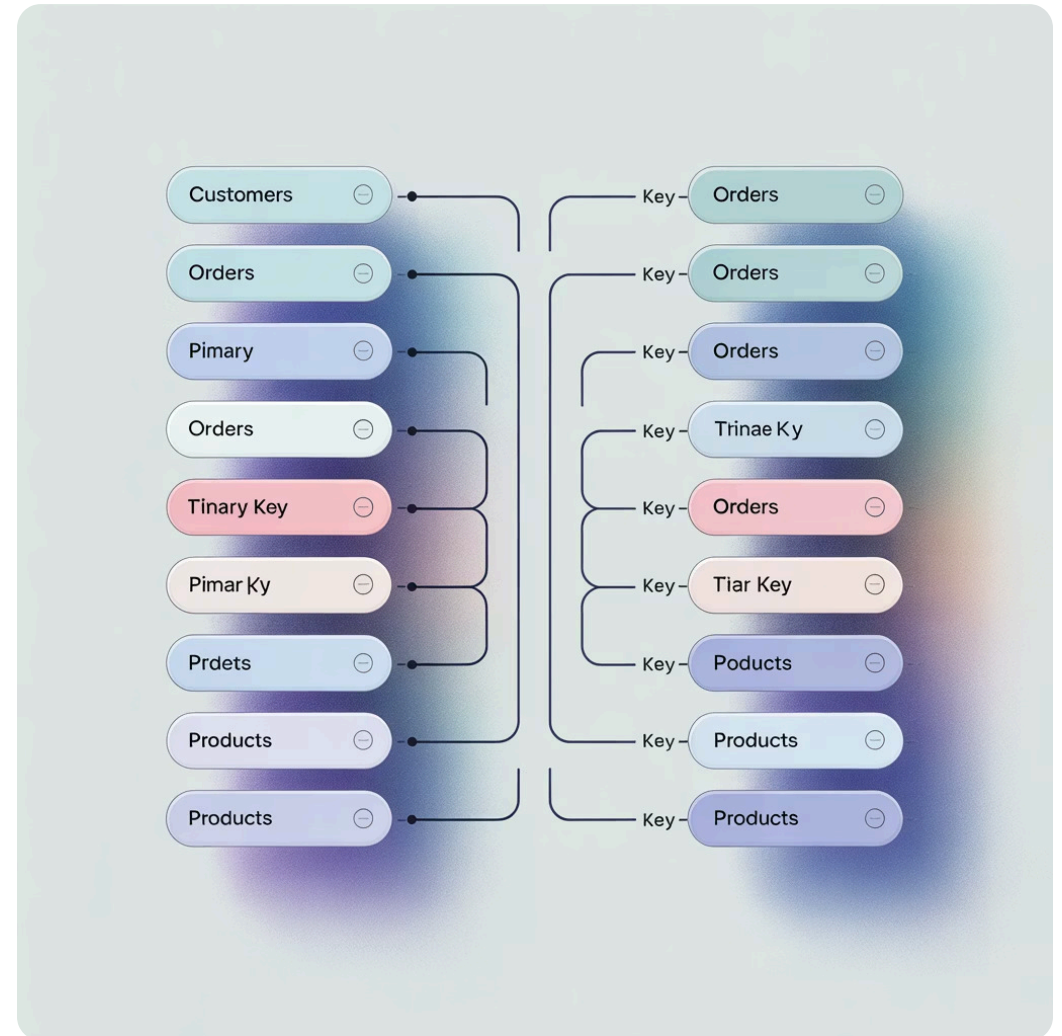
Una clave primaria es un atributo (o combinación de atributos) que identifica de forma única cada instancia de entidad dentro de un conjunto de entidades.

Características

- Debe ser único para cada instancia de entidad
- No puede contener valores nulos
- Debe ser mínimo (sin atributos innecesarios)
- Debe ser estable (poco probable que cambie)

Ejemplos

- Estudiante: ID_Estudiante
- Curso: Código_Curso
- Profesor: ID_Profesor



Las claves primarias son fundamentales para el diseño de bases de datos, ya que proporcionan la base para la identificación de entidades y el establecimiento de relaciones. Se convierten en la base de la estructura de la tabla en el modelo lógico.

Claves externas e integridad referencial

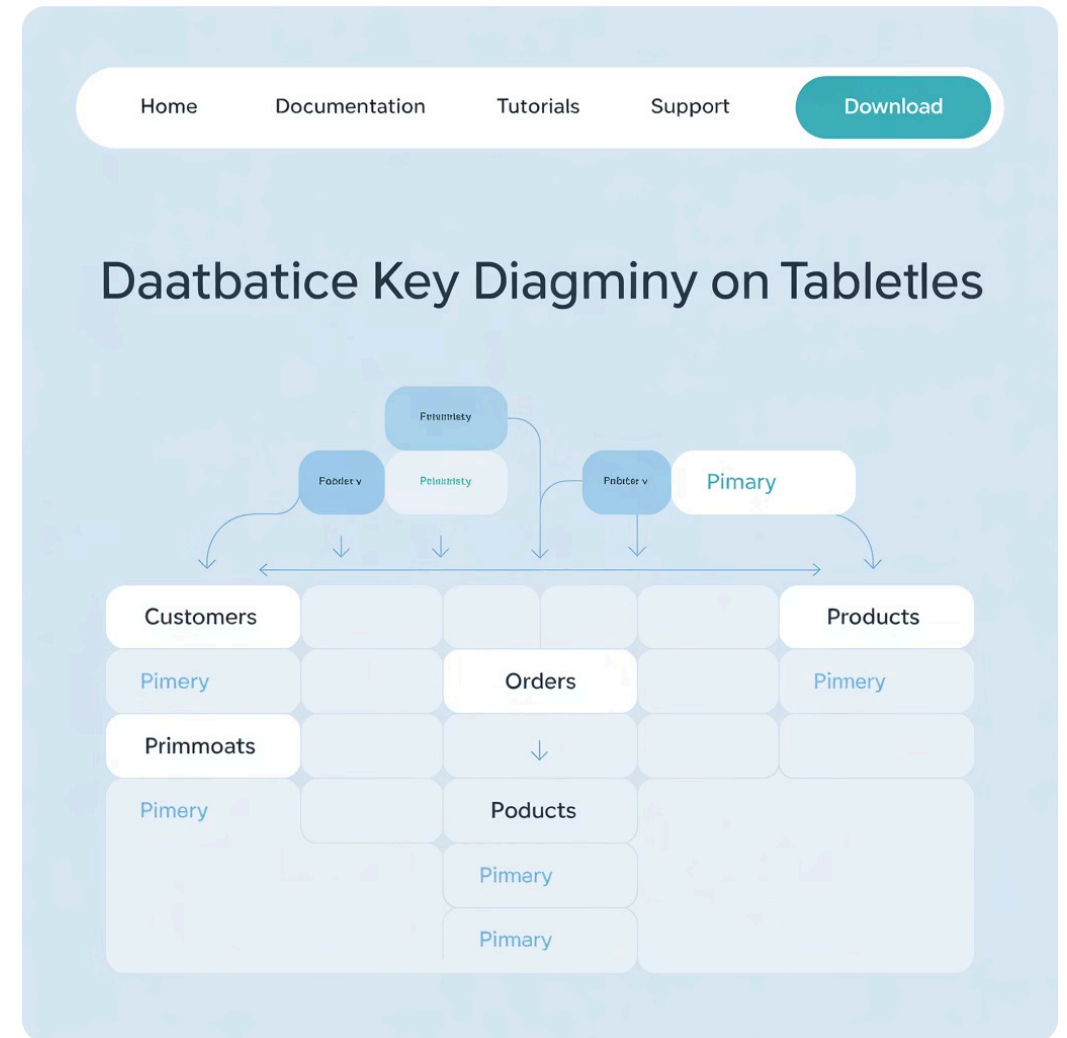
Claves externas

Una clave externa es un atributo en una entidad que se refiere a la clave principal de otra entidad, estableciendo una relación entre ellas.

- Crea conexiones lógicas entre entidades
- Implementa relaciones en el modelo lógico
- Puede ser nulo en algunos casos (relaciones opcionales)

Integridad referencial

Una restricción que garantiza que las relaciones entre tablas permanezcan consistentes. Cuando existe un valor de clave externa, debe referirse a un valor de clave principal existente en la tabla principal.



Sin integridad referencial, las bases de datos podrían contener registros huérfanos o relaciones no válidas que comprometan la consistencia y confiabilidad de los datos.

Acciones de Integridad Referencial

Cascada

Cuando se actualiza o elimina un registro en la tabla principal, se aplica automáticamente la misma acción (actualización o eliminación) a los registros relacionados en la tabla secundaria que contienen la clave externa.

Ejemplo: Si se cambia el nombre de un departamento, actualiza todos los registros de los empleados para que reflejen el nuevo nombre del departamento.

Restringir

Evita la eliminación o actualización de un valor de clave principal si los valores de clave externa de otras tablas lo referencian.

Ejemplo: Evita la eliminación de un curso si hay estudiantes inscritos actualmente en él.

Establecer nulo

Cuando se elimina o actualiza la clave principal, establece los valores de clave externa que la referencian en NULL.

Ejemplo: Si un profesor se va, establece el campo del asesor para sus estudiantes en NULL.

Establecer valor predeterminado

Cuando se elimina o actualiza la clave principal, establece los valores de clave externa que la referencian en un valor predeterminado.

Ejemplo: Si se disuelve un departamento, asigna a los empleados a un departamento predeterminado "Sin asignar".

Estas acciones determinan cómo la base de datos mantiene la integridad referencial cuando se producen cambios en los datos relacionados.

Entidades Débiles

Definición

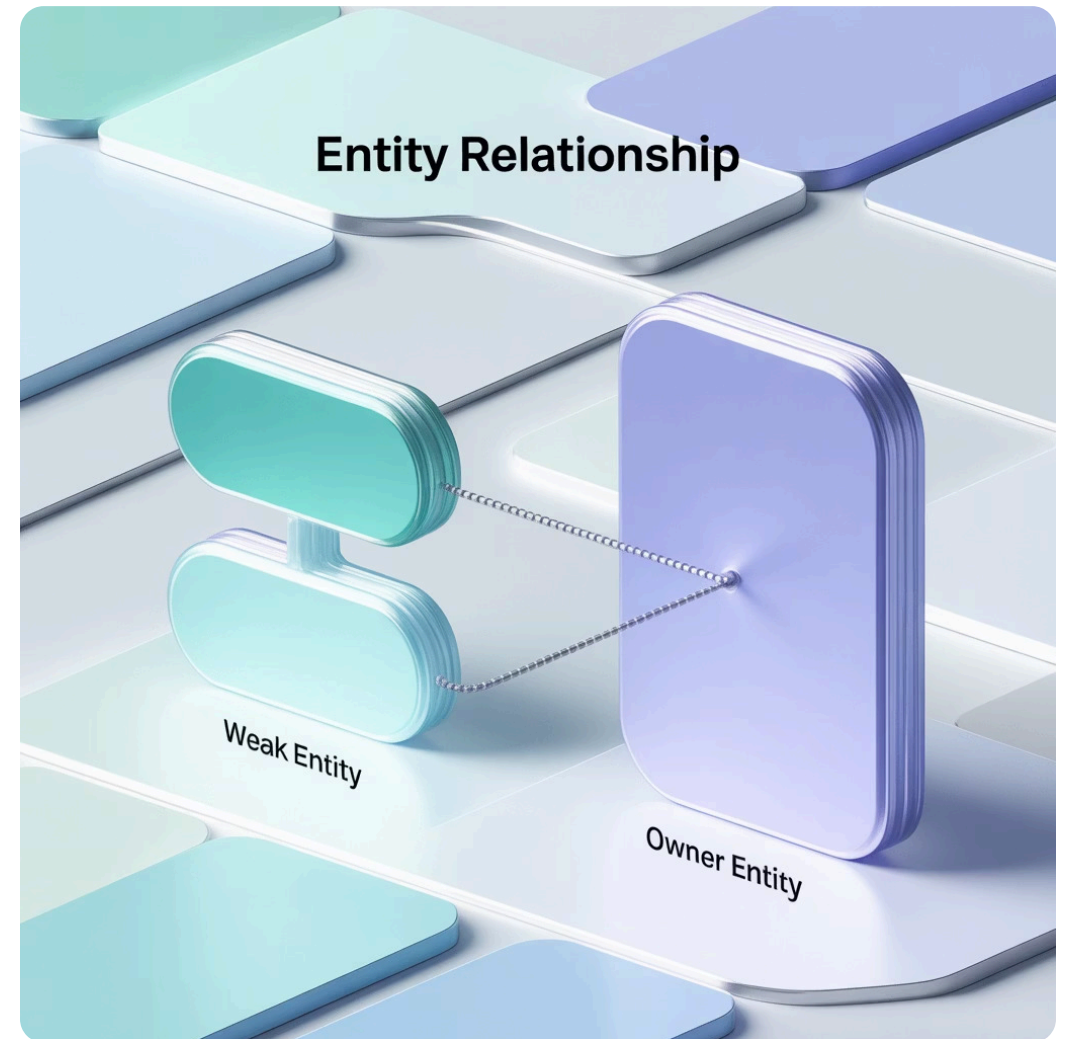
Una entidad débil es una entidad que no puede ser identificada de forma única solo por sus atributos y depende de la relación con otra entidad (entidad propietaria).

Características

- Depende de una entidad propietaria para su identificación
- Tiene una clave parcial (discriminador) que la identifica dentro del contexto de su propietario
- La existencia depende de la entidad propietaria (dependencia de existencia)
- Representada con rectángulos dobles en los diagramas ER

Ejemplo

Una entidad "Examen" podría ser débil si se identifica tanto por un ID de curso (de la entidad Curso) como por un número de examen (su clave parcial).



Las entidades débiles son importantes para modelar escenarios donde algunos objetos de datos no tienen existencia o identificación independiente fuera de su relación con otra entidad.

Del modelo conceptual al modelo lógico

Identificar entidades y atributos

Defina los objetos principales y sus propiedades a partir del análisis de requisitos.

Establecer relaciones

Determine cómo interactúan las entidades y la naturaleza de sus asociaciones.

Crear el diagrama ER

Visualice el modelo conceptual utilizando notación estandarizada.

Definir claves primarias y externas

Establezca la identificación de la entidad y la implementación de la relación.

Resolver relaciones de muchos a muchos

Transforme las relaciones M:N en dos relaciones 1:N con una entidad de intersección.

Crear esquema relacional

Convierta el modelo ER refinado en tablas, columnas y restricciones.

Este proceso transforma un modelo conceptual abstracto en un modelo lógico estructurado que se puede implementar en un sistema de gestión de bases de datos.

Características extendidas de ER

Especialización y generalización

Representa las relaciones de herencia entre los tipos de entidades (entidades de superclase y subclase).

Ejemplo: Persona (superclase) con Estudiante y Empleado como subclases.

Agregación

Representa una relación de "consta de" donde una entidad representa una colección de otras instancias de entidad.

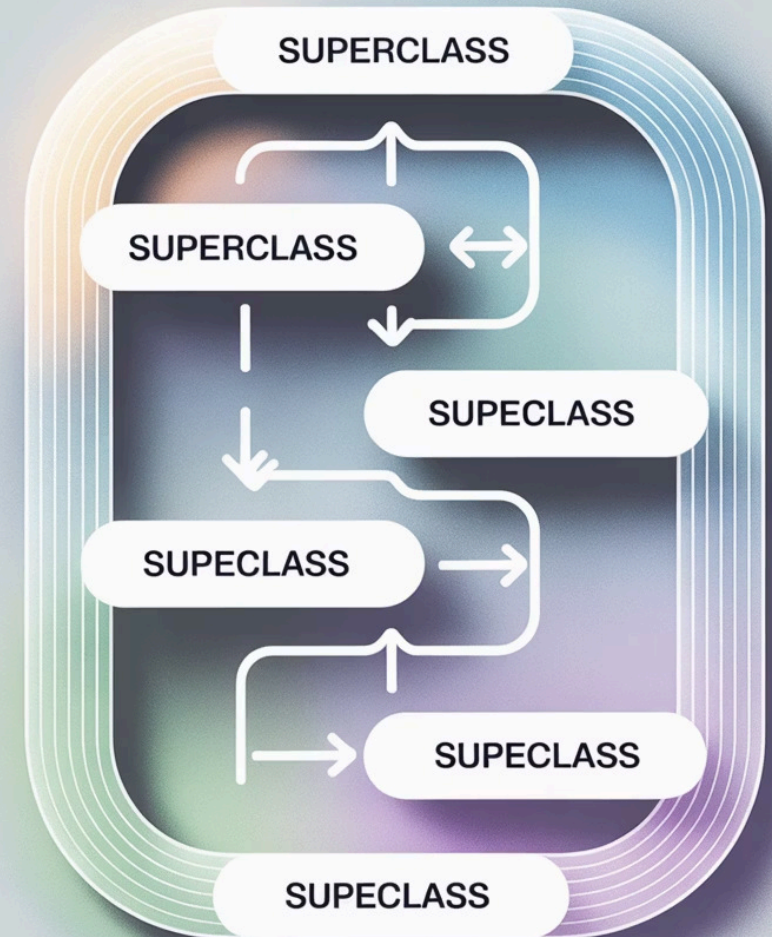
Ejemplo: Una entidad Equipo que consta de varias entidades Persona.

Relaciones ternarias

Relaciones que involucran a tres entidades simultáneamente.

Ejemplo: Estudiante, Curso y Semestre en una relación de "Inscripción".

Estas características extendidas proporcionan una potencia de modelado adicional para escenarios complejos del mundo real que no se pueden representar adecuadamente con construcciones ER básicas.

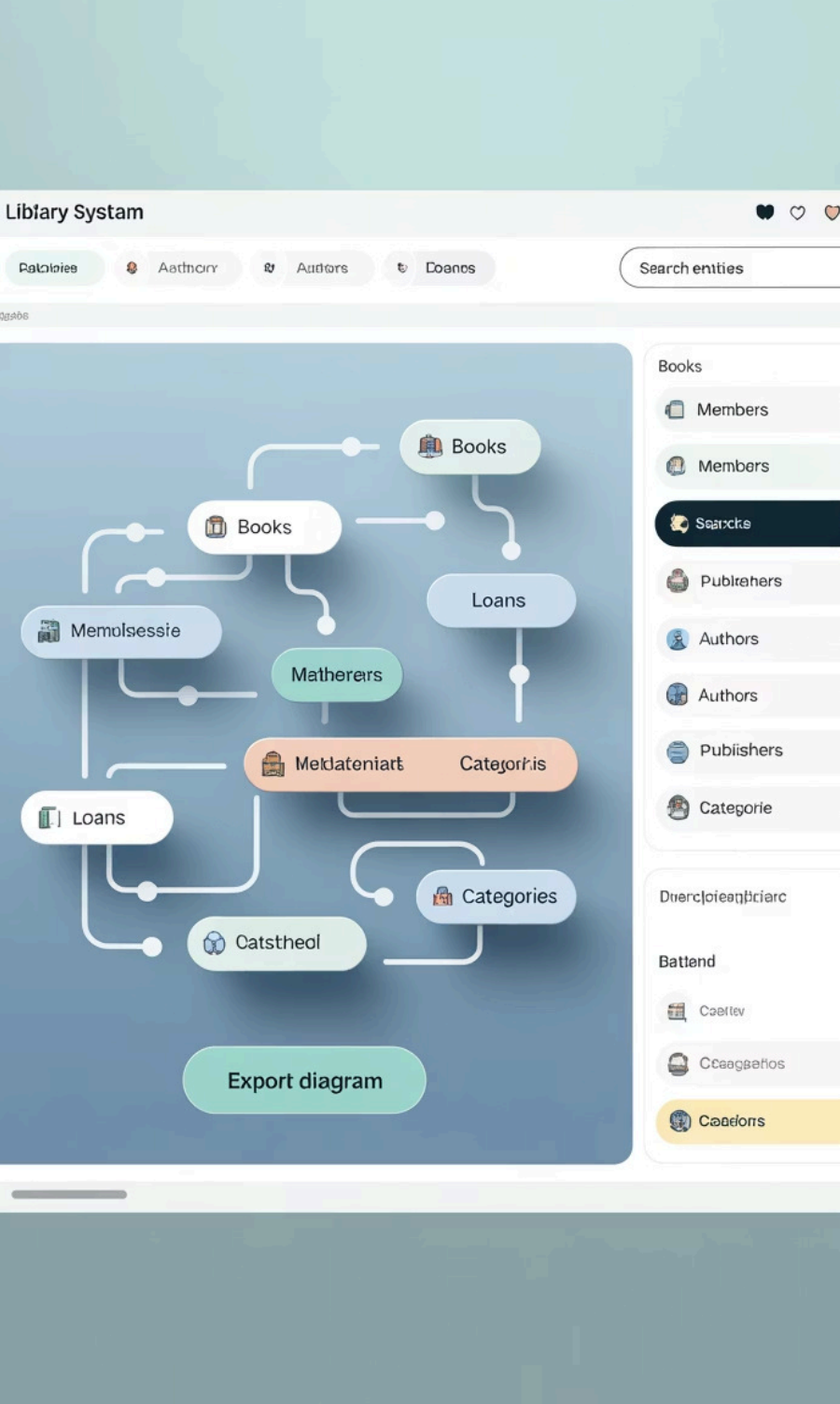


Actividad práctica: Sistema de biblioteca

Requisitos de diseño

Cree un diagrama de entidad-relación completo para un sistema de gestión de biblioteca con las siguientes especificaciones:

- Identifique las entidades principales (libros, miembros, préstamos, etc.)
- Defina los atributos apropiados para cada entidad (incluidas las claves primarias)
- Establezca relaciones entre entidades con la cardinalidad adecuada
- Considere entidades especializadas como autor, editorial, categoría
- Defina restricciones de integridad para garantizar la coherencia de los datos



Sistema de biblioteca: Solución de muestra

Entidades y atributos

- **Libro** (ISBN, Título, Año_de_publicación, Descripción, Copias_disponibles)
- **Miembro** (ID_de_miembro, Nombre, Dirección, Teléfono, Correo_electrónico, Fecha_de_incorporación)
- **Préstamo** (ID_de_préstamo, Fecha_de_préstamo, Fecha_de_vencimiento, Fecha_de_devolución, Importe_de_multa)
- **Autor** (ID_de_autor, Nombre, Biografía)
- **Editorial** (ID_de_editorial, Nombre, Dirección)
- **Categoría** (ID_de_categoría, Nombre, Descripción)

Relaciones

- **Libro** "escrito por" **Autor** (N:M)
- **Libro** "publicado por" **Editorial** (N:1)
- **Libro** "pertenece a" **Categoría** (N:M)
- **Miembro** "toma prestado" **Libro** a través de **Préstamo** (1:N)
- **Libro** "prestado en" **Préstamo** (1:N)

Tenga en cuenta que las relaciones de muchos a muchos requerirían entidades de intersección en el modelo lógico.

Errores comunes en el modelado ER

Confundir entidades y atributos

Modelar incorrectamente lo que debería ser una entidad como un atributo o viceversa. Por ejemplo, modelar "Autor" como un atributo de "Libro" en lugar de como una entidad separada con sus propios atributos.

Relaciones redundantes

Crear relaciones directas que pueden derivarse de las existentes. Por ejemplo, si A se relaciona con B y B se relaciona con C, una relación directa de A a C podría ser redundante.

Cardinalidad incorrecta de la relación

Identificar erróneamente la naturaleza de las relaciones (1:1, 1:N, N:M). Por ejemplo, modelar "Estudiante-Curso" como 1:N cuando debería ser N:M.

Ignorar las reglas de negocio

No incorporar restricciones o reglas de negocio importantes en el modelo, lo que lleva a diseños que no pueden garantizar la integridad de los datos.

Evitar estos errores comunes conducirá a modelos de bases de datos más claros, precisos y útiles.



Resumen

Lo que hemos aprendido

- El propósito y la importancia del modelado conceptual
- Componentes centrales del modelo Entidad-Relación
- Tipos de relaciones y su cardinalidad
- Creación e interpretación de diagramas ER
- Claves primarias, claves externas e integridad referencial
- Características ER extendidas para escenarios complejos