



Fundamentos de Bases de Datos

Fundamentos de Bases de Datos

Indice

- Bienvenida, conceptos básicos y Contexto Histórico de las Bases de Datos
- Introducción a las Bases de Datos Relacionales
 - Historia de las RDB
 - Entidades y Atributos
 - Entidades de Platzi Blog
 - Relaciones
 - Múltiples Muchos
 - Diagrama Entidad-Relación

Indice

- [Módulo 1: Bienvenida, conceptos básicos y Contexto Histórico de las Bases de Datos](#)
- [Módulo 2: Introducción a las Bases de Datos Relacionales](#)
 - [Historia de las RDB](#)
 - [Entidades y Atributos](#)
 - [Entidades de Platzi Blog](#)
 - [Relaciones](#)
 - [Múltiples Muchos](#)
 - [Diagrama Entidad-Relación](#)
 - Diagrama Físico: Tipos de Datos y Constraints
 - Diagrama Físico: Normalizando PlatziBlog
 - Formas Normales en DB Relacionales
- Módulo 3: RDBMS (MySQL) o Cómo Hacer lo Anterior de Manera Práctica
 - RDB ¿Qué?
 - Instalación Local de un RDBMS (Windows)
 - Instalación Local de un RDBMS (Mac)
 - Instalación Local de un RDBMS (Ubuntu)
 - Clientes Gráficos
 - Servicios Administradores
- Módulo 4: SQL Hasta en la Sopa
 - Historia de SQL
 - DDL Create
 - Create View y DDL Alter
 - DDL Drop

- DML
- ¿Qué tan Standard es SQL?
- Creando Platziblog: Tablas Independientes
- Creando Platziblog: Tablas Dependientes
- Creando Platziblog: Tablas Transitivas
- Módulo 5: Consultas a una Base de Datos
 - ¿Por Qué las Consultas son tan Importantes?
 - Estructura Básica de un Query
 - SELECT
 - FROM
 - Utilizando la Sentencia FROM
 - WHERE
 - Utilizando la Sentencia WHERE Nulo y no Nulo
 - GROUP BY
 - ORDER BY y HAVING
 - El Interminable Agujero de Conejo (Nested Queries)
 - ¿Cómo Convertir Una Pregunta en un Query SQL?
 - Preguntándole a la Base de Datos
 - Consultando PlatziBlog
- Módulo 6: Introducción a las Bases de Datos NO Relacionales
 - ¿Qué son y Cuáles son los Tipos de Bases de Datos no Relacionales?
 - Servicios Administrados y Jerarquía de Datos
- Módulo 7: Manejo de Modelos de Datos en Bases de Datos no Relacionales
 - Top Level Collection con Firebase
 - Creando y Borrando Documentos en Firestore
 - Colecciones vs Subcolecciones
 - Recreando Platziblog
 - Construyendo Platziblog en Firestore
 - Proyecto Final: Transformando tu Proyecto en una DB no Relacional
- Módulo 8: Bases de Datos en la Vida Real
 - Bases de Datos en la Vida Real
 - Big Data
 - Data Warehouse
 - Data Mining
 - ETL
 - Business Intelligence
 - Machine Learning
 - Data Science
 - ¿Por Qué Aprender Bases de Datos Hoy?
- Bonus
 - Bases de Datos Relacionales vs no Relacionales
 - Elegir una Base de Datos

Bienvenida, conceptos básicos y Contexto Histórico de las Bases de Datos

Profesor: [Israel Vázquez](#).

¿Qué son las bases de datos? Complementan la arquitectura de Von Neumann.

Se dividen en dos:

Relacionales: SQL Server (Microsoft), SQL (Oracle). Versiones Open Source: PostgreSQL, neo4j, MariaDB.

No relacionales: MemcacheDB, Cassandra (Facebook), DynamoDB, elasticsearch, BigQuery, MongoDB.

Autoadministrados: Se instala en mi PC/Servidor, me encargo de actualizaciones, mantenimiento, etc.

Administrados: Servicios que se ofertan. No se instala ni hace mantenimiento.

Introducción a las Bases de Datos Relacionales

Historia de las RDB

Surgen de la necesidad de conservar la información más allá de la memoria RAM. En la arquitectura de Von Neumann se contempla la RAM y el procesamiento de datos, pero no el guardado de datos persistente (se apaga la PC, se inicia y se conservan los datos).

Una primera aproximación fueron las bases de datos basadas en archivos: guardar datos separados por comas (texto plano) fácil de guardar pero no de recuperar ordenadamente. Hay necesidad de algo más estructurado:

[Edgar Codd](#): Inventor de Bases de Datos Relacionales.

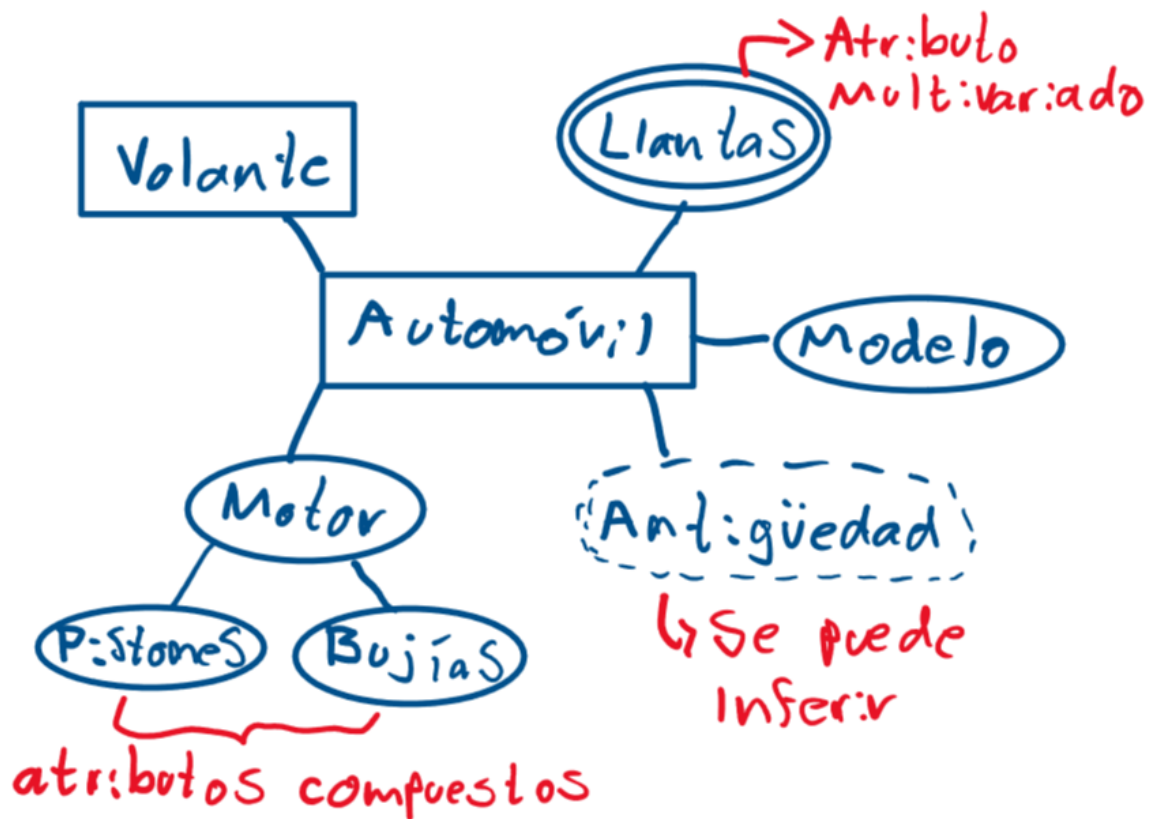
Inventó el [álgebra relacional](#). Cómo tenemos datos que se pueden empezar a mezclar y unir a través de diferentes propiedades y características.

[12 Reglas de Codd](#)¹:

Entidades y Atributos

Entidad: es similar a un **objeto**, que representa algo de la vida real.

Tienen **atributos** que son las cosas que la hacen ser una entidad.



Las entidades se escriben en **plural**. No. de serie (**dato llave**) ayuda a identificar de forma única a cada computadora dentro de la entidad/grupo de computadoras.

no de serie	color	año	pantalla
LKJ789JKAS	gris	2017	AX4829i
KCO3100KJH	negro	2019	AX4930i
NSDJOIH128	negro	2018	AX4930i
09KSIHBD71	gris	2017	AX4829i

Hay dos tipos de entidades. Las vistas hasta ahora son **entidades fuertes** (no dependen de ninguna otra entidad para existir). Las **entidades débiles** no pueden existir sin una entidad fuerte. Se representan con rectángulos pero con **doble línea**.

Pueden ser débiles por 2 motivos:

- **Identidad** (No se diferencian entre sí más que por la clave de su entidad fuerte)

Libros			Ejemplares		
id	título	...	libro_id	localización	edición
LKJ789JKAS	Viaje al cent...	...	LKJ789JKAS	pasillo 1	1
KCO3100KJH	El señor de	KCO3100KJH	pasillo 1	1
NSDJOIH128	De la tierra...	...	NSDJOIH128	pasillo 1	3
09KSIHBD71	Amor en tie...	...	09KSIHBD71	pasillo 1	1

Se les asigna el id del libro. Las hace dependientes al libro gracias al campo de identificación.

- **Existencia** (Aunque se agregue id diferente al de libro, aun así, conceptualmente no se puede tener un ejemplar sin un libro primero).

Libros			Ejemplares		
id	título	...	id	localización	edición
LKJ789JKAS	Viaje al cent...	...	JKE7823CLK	pasillo 1	1
KCO3100KJH	El señor de	JKFE1093JD	pasillo 1	1
NSDJOIH128	De la tierra...	...	82938ISHDIK	pasillo 1	3
09KSIHBD71	Amor en tie...	...	838439JHDUI	pasillo 1	1

Se les asigna id propio.

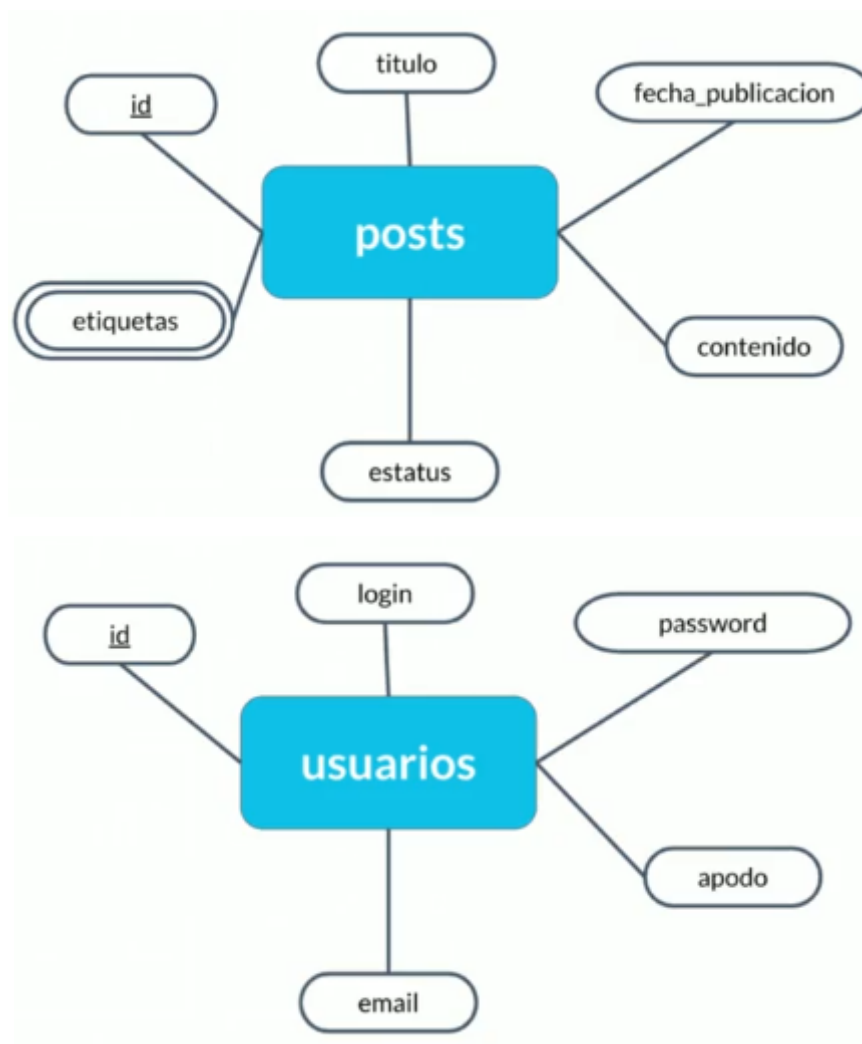
Entidades de Platzi Blog

Proyecto del curso: PlatziBlog.

Paso 1: **identificar entidades** (cualquier objeto que se pueda representar en un grupo)

- Posts
- Usuarios
- Comentarios
- Categorías

Paso 2: **Pensar en los atributos**



Relaciones

(Los diagramas que hemos hecho se llaman diagrama **entidad-relación**)

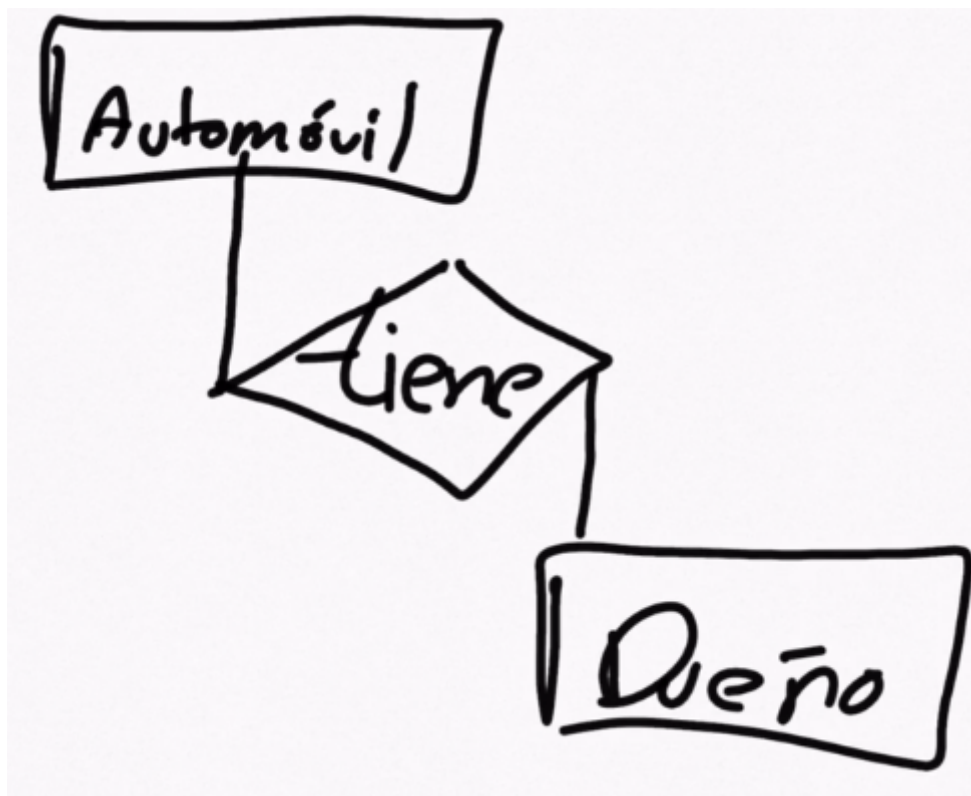
Relaciones: manera en la que empezamos a ligar diferentes entidades u objetos.

Se representan con un rombo.

Se definen a través de verbos.

Ejemplo:

"El dueño tiene un automóvil"

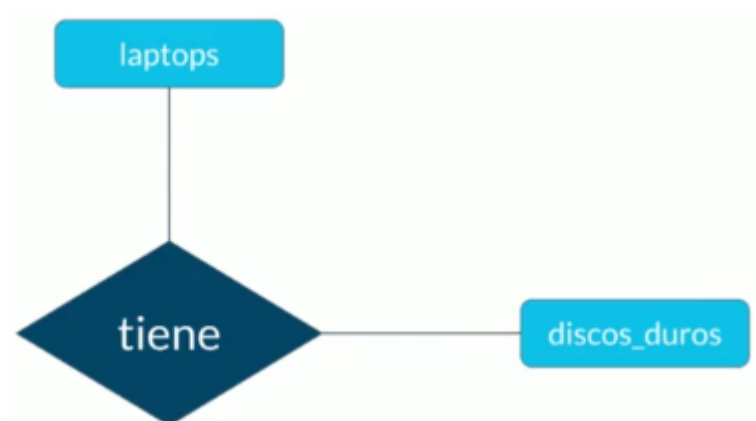


Otro ejemplo:

Relación entre jugadores y equipos.

Jugador -> pertenece a -> equipo

Otro ejemplo:



*discos_duros no era entidad, sino atributo multivaluado, por lo general se convierten en entidades separadas porque tienen diferencias entre sí y se relacionan de varias maneras con laptops.

Cómo se define esta relación de cuántos discos duros tiene la laptop?

Tiene que ver con una propiedad llamada **cardinalidad**

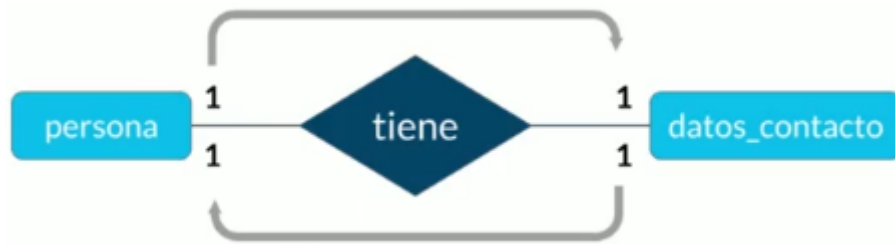
Hay diferentes tipos:

Cardinalidad 1 a 1:



La cardinalidad se define porque se hace el enunciado: "una persona cuántos datos de contacto tiene?" tiene **una serie** de datos de contacto.

Ahora del otro lado: "los datos de contacto (una serie) pertenecen a cuántas personas?" Solamente a **una**.



Para sacar la cardinalidad, se debe tomar el número mayor de ambos lados, como es 1 y 1, se tiene una **cardinalidad de 1:1**.

También hay otro tipo de diagramas en que esta relación se encuentra con este conector:



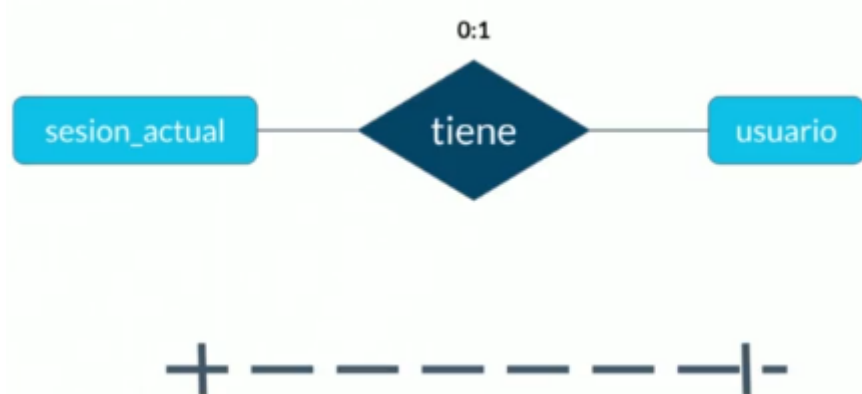
O más estricto (1 y solo 1):



Cardinalidad 0 a 1 o 1 a 1 opcional (puede haber la opción de que no exista ninguno de uno de los lados):

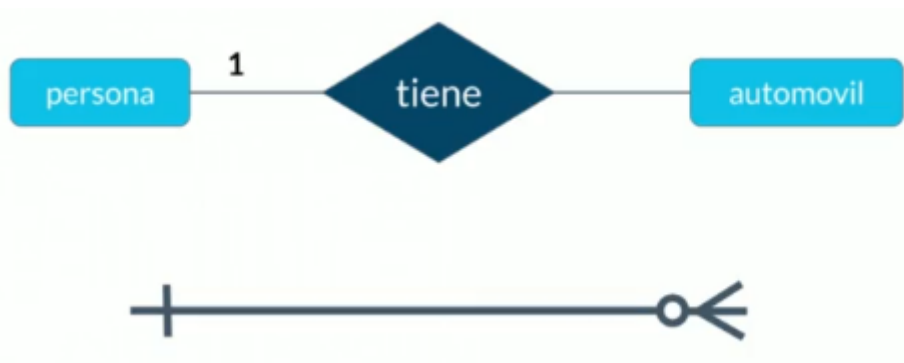


Ejemplo:

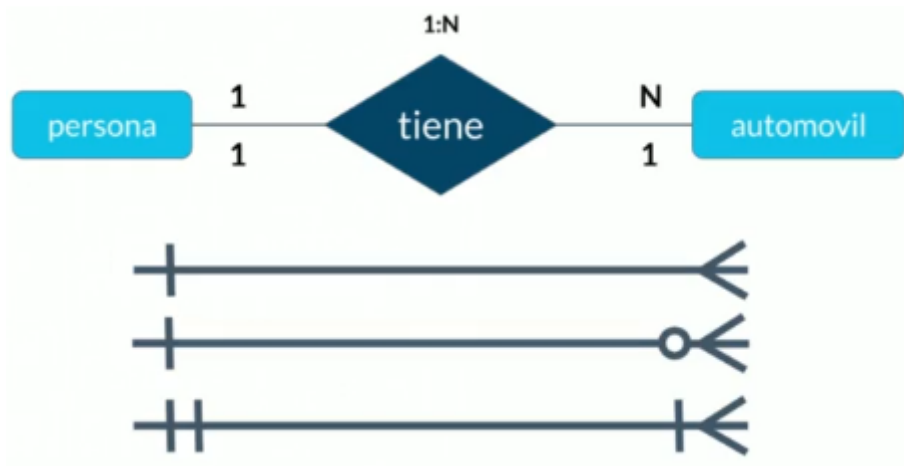


La sesión actual tiene que tener un usuario, pero un usuario **puede o no** estar en sesión en este momento. Por lo tanto, la cardinalidad se define de 0:1 porque es opcional. Se representa con la línea punteada.

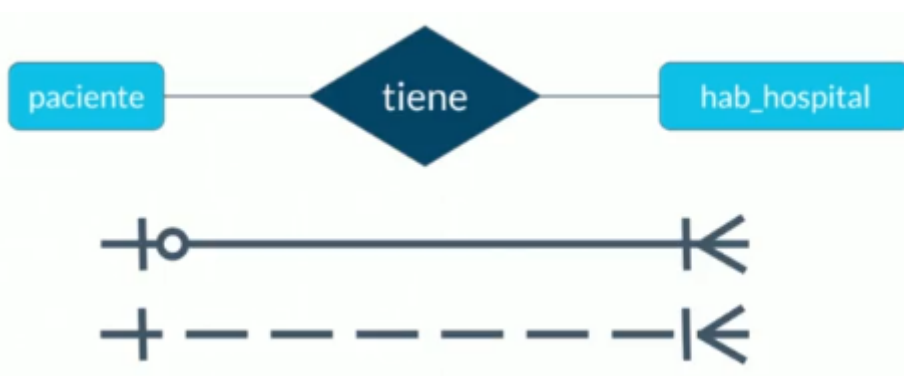
Cardinalidad 1 a N o 1 a muchos:



1 persona puede tener muchos automóviles. Cada automóvil solo puede pertenecer a una persona.



Cardinalidad 0 a N:



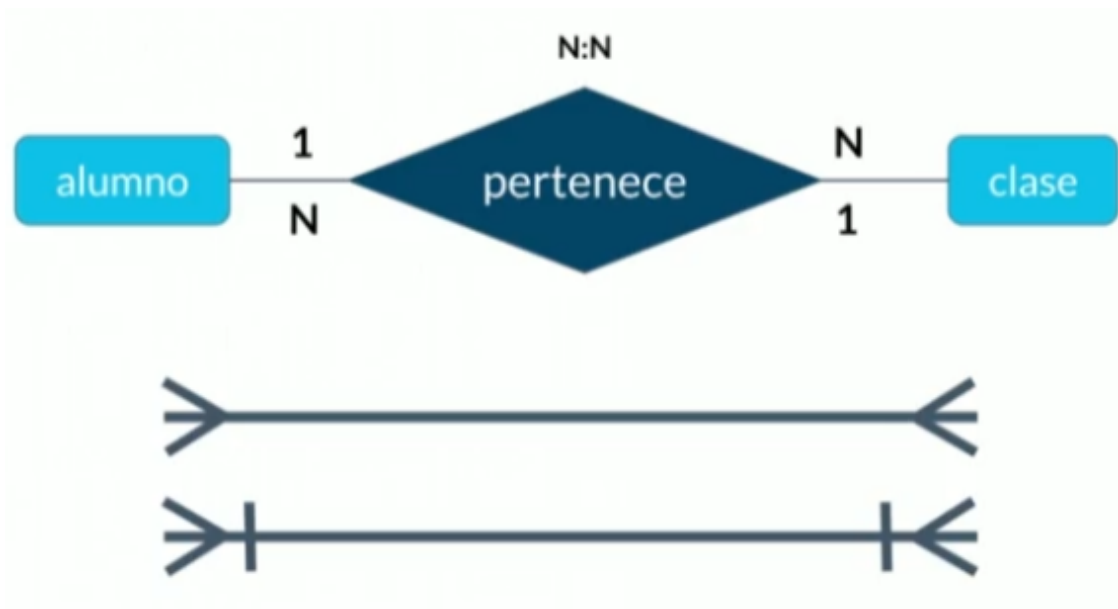
Un paciente siempre está asignado a una habitación de hospital. Pero una habitación, puede estar vacía.

Múltiples Muchos

Cardinalidad N a N:

Un alumno puede estar inscrito o tomar varias clases.

Una clase puede tener varios alumnos.



La versión estricta añade una línea cruzada.

Diagrama Entidad-Relación

1. <https://www.mindmeister.com/es/1079684487/las-12-reglas-de-codd-del-modelo-relacional?fullscreen=1#>