



Base de Datos

Modelado de BD



INTRODUCCIÓN

Modelado de datos

El modelado de datos es el proceso de representar y organizar la estructura y relaciones de los datos en un sistema de información mediante modelos de datos. El objetivo principal del modelado de datos es comprender y visualizar cómo se organizan los datos y cómo interactúan entre sí. Esto proporciona una base sólida para el diseño y desarrollo de bases de datos eficientes y precisas.

<input type="checkbox"/>	carro_id	tipo	color	tienda	precio
<input type="checkbox"/>	1	5	negro	1	203000
<input type="checkbox"/>	2	2	amarillo	1	150
<input type="checkbox"/>	3	3	verde	3	14800
<input type="checkbox"/>	4	4	negro	2	10200
<input type="checkbox"/>	5	1	verde	4	82399
<input type="checkbox"/>	6	1	azul	4	22374

<input type="checkbox"/>	tipo_id	marca_id	nombre
<input type="checkbox"/>	1	3	tipo e
<input type="checkbox"/>	2	2	ll118
<input type="checkbox"/>	3	4	golf
<input type="checkbox"/>	4	4	polo
<input type="checkbox"/>	5	1	quattroporte

<input type="checkbox"/>	marca_id	nombre	pais_de_origen
<input type="checkbox"/>	1	maserati	italia
<input type="checkbox"/>	2	lada	rusia
<input type="checkbox"/>	3	jaguar	reino unido
<input type="checkbox"/>	4	volkswagen	alemania

<input type="checkbox"/>	tienda_id	nombre	calle	codigo_postal	telefono
<input type="checkbox"/>	1	amsterdam south	churchill	1079BA	20373
<input type="checkbox"/>	2	amsterdam west	mercator	1056RT	20838
<input type="checkbox"/>	3	amsterdam north	buikslote	1031AB	20387
<input type="checkbox"/>	4	the hague	neherstrem	2491JJ	70387

En esta imagen vemos la representación de una Base de Datos y te mencionamos algunas cosas importantes:

- Cada **base de datos** se compone de **una o más tablas** que guarda un conjunto de datos.
- Cada **tabla** tiene una o más **columnas** y **filas**.
- Las **columnas** guardan una **parte de la información** sobre cada elemento que queremos guardar en la tabla, *cada fila de la tabla conforma un registro*.

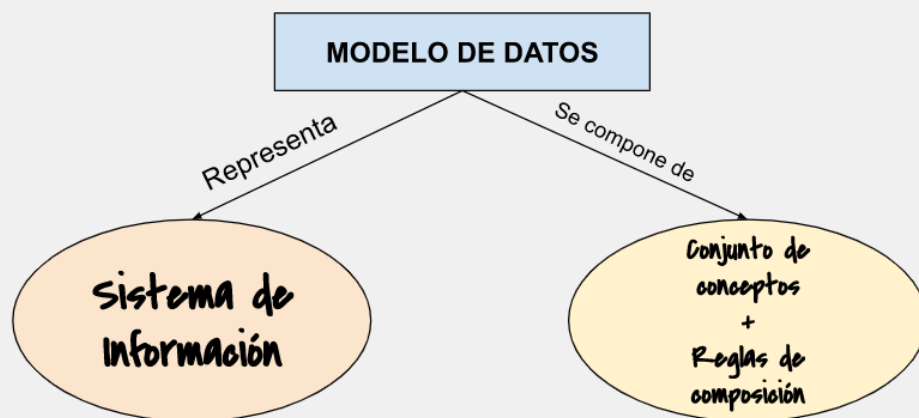
No importa si nuestra base de datos tiene sólo 20 registros, o algunos cuantos miles, es importante asegurarnos que nuestra base de datos está correctamente diseñada para que tenga eficiencia y que se pueda seguir utilizando por largo del tiempo.

El **diseño de bases de datos** es el **proceso por el que se determina la organización de una base de datos**, incluidos su estructura, contenido y las aplicaciones que se han de desarrollar.

Para ello necesitamos conocer sobre Modelos de Datos

¿Qué es un modelo de Datos?

Es muy importante seleccionar el modelo de datos a utilizar, ya que ayuda a los equipos de desarrollo de software a *diseñar* y *organizar* datos asociados al *proyecto de software*.



Para comprender los tipos de Modelo de Datos debemos empezar sabiendo que el diseño de una base de datos se descompone en **diseño conceptual**, **diseño lógico** y **diseño físico**.

Diseño conceptual

Parte de las especificaciones de requisitos de usuario y su resultado es el esquema conceptual de la base de datos.

¿Qué es un esquema conceptual? es una descripción de alto nivel de la estructura de la base de datos, independientemente del DBMS que se vaya a utilizar para manipularla.

El objetivo del diseño conceptual es describir los datos de la base de datos y no las estructuras de almacenamiento que se necesitarán para manejar estos datos.

Diseño lógico

Parte del esquema conceptual y da como resultado un esquema lógico.

¿Qué es un esquema lógico? Es una descripción de la estructura de la base de datos en términos de las estructuras de datos que puede procesar un tipo de DBMS.

Un modelo lógico es un lenguaje usado para especificar esquemas lógicos (modelo relacional, modelo de red, etc.).

El diseño lógico depende del tipo de DBMS que se vaya a utilizar, no depende del producto concreto.

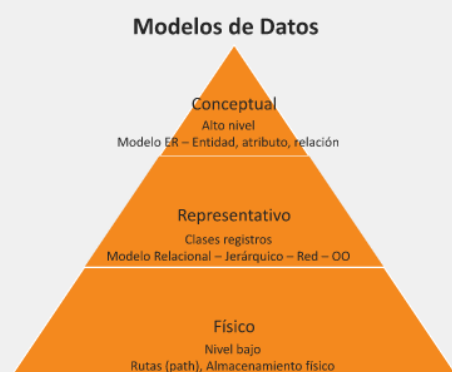


Diseño lógico

Parte del esquema lógico y da como resultado un esquema físico.

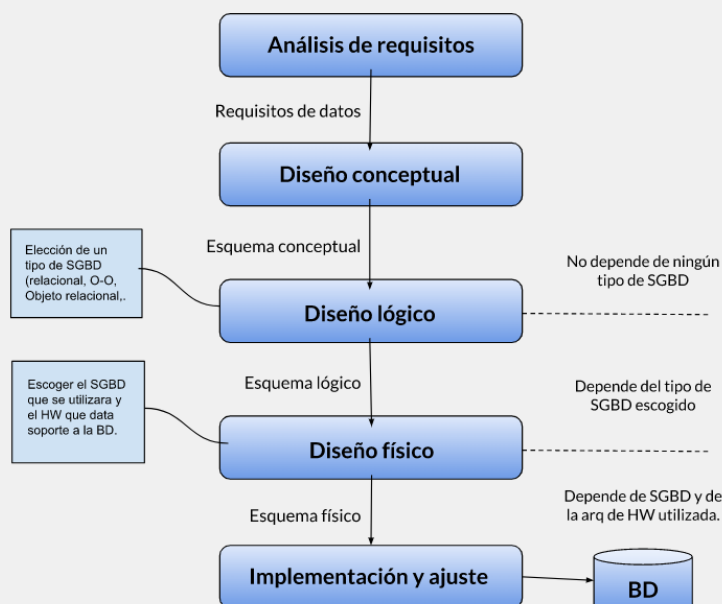
¿Qué es un esquema físico? es una descripción de la implementación de una base de datos en memoria secundaria: las estructuras de almacenamiento y los métodos utilizados para tener un acceso eficiente a los datos.

Por ello, el diseño físico depende del DBMS.



Proceso de Diseño de la Base de datos

Se muestra a continuación una descripción simplificada del proceso de diseño de la base de datos.





1. Obtención y análisis de requisitos.

Durante este paso los **diseñadores entrevistan a los futuros usuarios** de la base de datos para entender y documentar sus requisitos de datos.

El resultado de este paso será un **conjunto de requisitos del usuario**, redactado de forma concisa. Estos requisitos deben especificarse de la forma más detallada y completa posible.

2. Esquema conceptual.

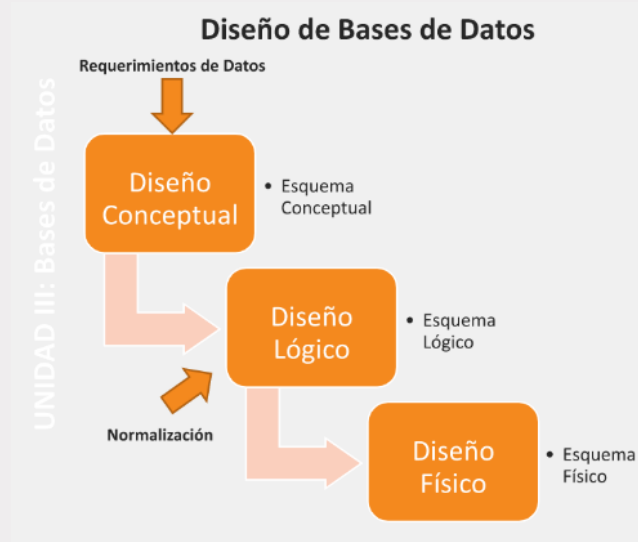
Se crea un **esquema conceptual** para la base de datos mediante un modelo conceptual de datos de alto nivel. Este paso se denomina **diseño conceptual**. El esquema conceptual es una **descripción concisa de los requisitos de información de los usuarios**, y contiene descripciones detalladas de los tipos de **entidad, vínculos y restricciones representados** según el modelo conceptual de datos usado. Puestos que estos conceptos no incluyen detalles de implementación, suelen ser **fáciles de entender y pueden servir para comunicarse con usuarios no técnicos**.

El esquema conceptual de alto nivel también puede servir como referencia para asegurarse de satisfacer todos los requerimientos de los usuarios y de que no haya conflictos entre dichos requisitos.

3. Una vez diseñado el esquema conceptual o durante dicho proceso, es posible utilizar las operaciones básicas del modelo de datos para especificar operaciones de usuario de alto nivel identificadas durante el análisis funcional.

A partir de allí se debe usar un SGBD para implementar la base de datos. Esto se logra **transformando el esquema conceptual del modelo** usado al **modelo de datos de implementación**. Este paso se llama **diseño lógico** o transformación del modelo de datos, y su resultado es un esquema de la base de datos en el modelo de datos que se usará para la implementación de la base de datos.

4. El paso final es la fase de **diseño físico**, durante la cual se especifican las **estructuras de almacenamiento internas, los caminos de acceso y la organización de ficheros de la base de datos**. Por ejemplo, en el curso desarrollaremos una aplicación y utilizaremos un SGBD para trabajar con los datos (MySQL Workbench).





MER

MER - Modelo Entidad Relación

Denominado por sus siglas como: MER, este modelo representa a la realidad a través de **entidades**, que son objetos que existen y que se distinguen de otros por sus características.



¿Cuáles son los elementos de un modelo?

1. Entidad
2. Atributos
3. Relación
 - a. Grado
 - b. Cardinalidad
 - c. Atributos propios de una relación
 - d. Clave de una relación
4. Claves



ENTIDAD

Una entidad es un **objeto del mundo real** que puede distinguirse de otros objetos.

Una entidad puede ser un **objeto con existencia física** (una persona, un automóvil, una casa, un empleado) o un **objeto con existencia conceptual** (una empresa, un puesto de trabajo, un curso universitario, una cuenta de cliente).

ATRIBUTO

Cada entidad tiene propiedades o características específicas llamadas **atributos que la describen**.

Por ejemplo, una entidad Cliente puede describirse por su nombre, fecha de nacimiento, dirección, dni y monto adeudado.

➤ Atributos compuestos o simples

Los **atributos compuestos** se pueden dividir en componentes más pequeños. Por ejemplo, para una **entidad Cliente** el atributo **Dirección** en realidad puede ofrecer información sobre Calle, Número, Localidad, etc.

En cambio, los **atributos simples** no pueden subdividirse, por ejemplo, Nombre, Edad.

➤ Atributos monovaluados o multivaluados

Los **atributos monovaluados** son aquellos que pueden tener un único valor por entidad, por ejemplo, para la **entidad Cliente** el atributo Género es monovaluado.

En cambio, un atributo que puede tener un conjunto de valores para la misma entidad es un atributo multivaluado, por ejemplo, para la **entidad Cliente** el atributo Profesión, ya que la persona puede tener varias Profesiones.

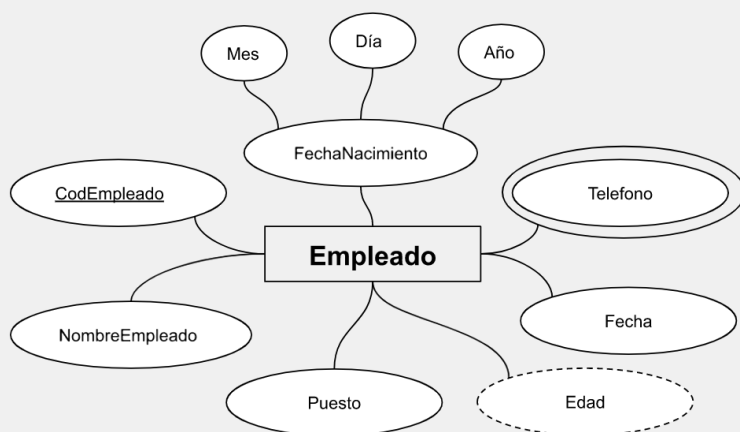
➤ Atributos almacenados o derivados

En algunos casos, los valores de dos o más atributos están relacionados, por ejemplo, los atributos Edad y Fecha de Nacimiento de una persona. Para un registro particular, el valor de la Edad puede estar determinado por la fecha actual y la fecha de nacimiento de la persona. Entonces el atributo Edad recibe el nombre de **atributo derivado** y se dice que es derivado del atributo Fecha de Nacimiento, el cual recibirá el nombre de **atributo almacenado**.



Además, para cada conjunto de entidades se escoge un **atributo clave**.

Una **clave** es un conjunto mínimo de atributos cuyos valores identifican de manera unívoca a cada entidad del conjunto. Puede haber más de una clave candidata; en ese caso, se escogerá una de ellas como clave principal.



Gráficamente, vamos a utilizar la notación de Chen para representar los MER.

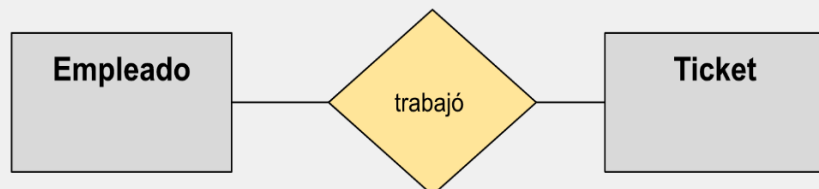
En rectángulos representaremos las entidades y en elipses los atributos.

- Los **atributos compuestos** se van a representar incluyendo los atributos simples en los que se descomponen, como el caso de *FechaNacimiento*.
- Los **atributos multivaluados** se van a representar con elipses dobles, como el caso de *Telefono* (ya que cada empleado puede tener más de uno).
- Los **atributos derivados o calculados** como el ejemplo de la *edad*, se van a representar con elipses de línea punteada.
- Los **atributos claves** serán subrayados en el diagrama, como es en este caso *CodEmpleado*.

Relación

Una relación es una **asociación entre dos o más entidades**.

Por ejemplo, puede que se tenga que modelar una relación en la cual un empleado haya trabajado en un ticket determinado.



En el conjunto de relaciones **trabajó**, cada relación indica un trabajo efectuado por ese empleado. Observá que puede que varios conjuntos de relaciones impliquen a los mismos conjuntos de entidades.

Veremos ahora las partes que conforman la relaciones

a) Grado

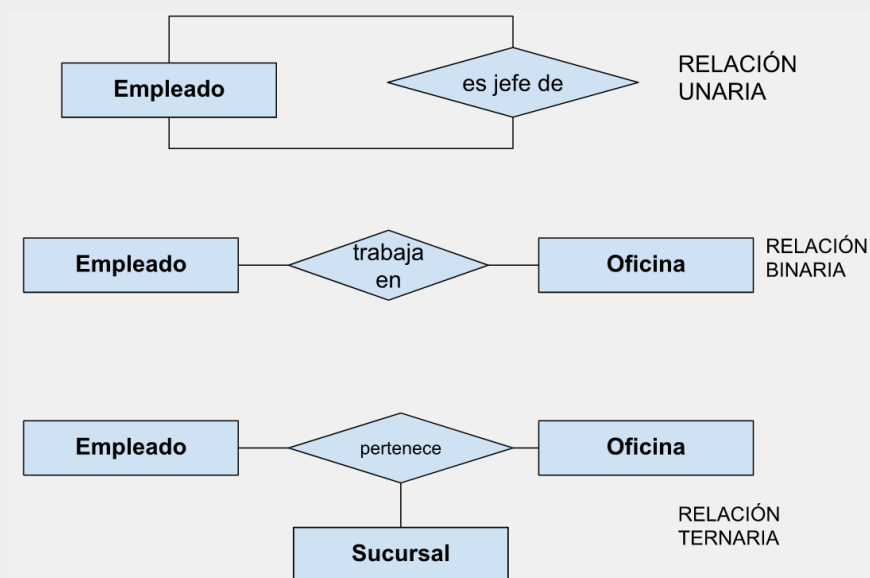
Se refiere al número de entidades que participan en una relación.

Los conjuntos de relaciones que involucran dos conjuntos de entidades se llaman relaciones binarias (o de grado dos). La mayoría de las relaciones en una base de datos es de este tipo.

Los conjuntos de relaciones pueden involucrar a más de dos conjuntos de entidades.

Nombre de la Relación	Grado
Unarias o Unitarias o de Reflexión	1
Binarias	2
Ternarias	3
N-arias	N

Algunos ejemplos de esto serían:





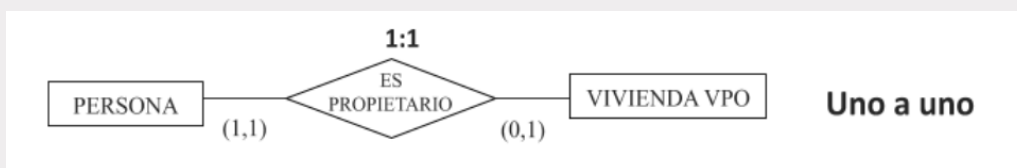
b) Cardinalidad

Dado un conjunto de relaciones en el que participan dos o más conjuntos de entidades, la cardinalidad de la correspondencia indica el número de entidades con las que puede estar relacionada una entidad dada.

Dado un conjunto de relaciones binarias y los conjuntos de entidades A y B, las cardinalidades pueden ser:

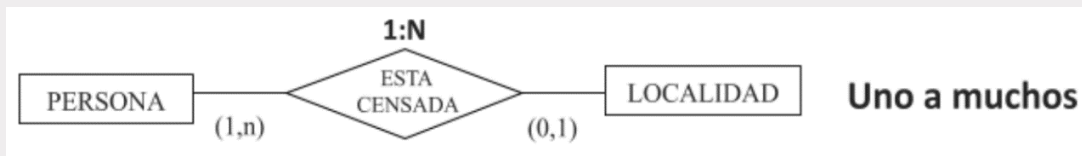
- **Uno a Uno: (1:1)**

Un registro de una entidad A se relaciona con solo un registro en una entidad B.



- **Uno a Varios: (1:N)**

Un registro en una entidad en A se relaciona con cero o muchos registros en una entidad B. Pero los registros de B solamente se relacionan con un registro en A.

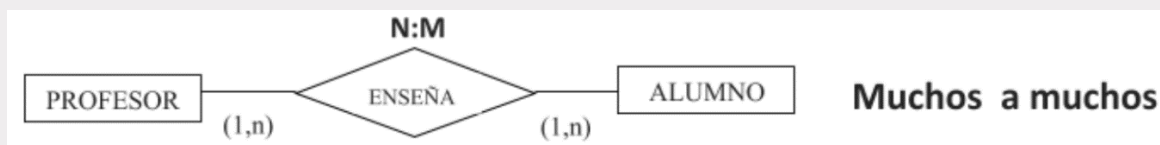


- **Varios a Uno: (N:1)**

Una entidad en A se relaciona exclusivamente con una entidad en B. Pero una entidad en B se puede relacionar con 0 o muchas entidades en A

- **Varios a Varios: (N:M)**

Una entidad en A se puede relacionar con 0 o con muchas entidades en B y viceversa.

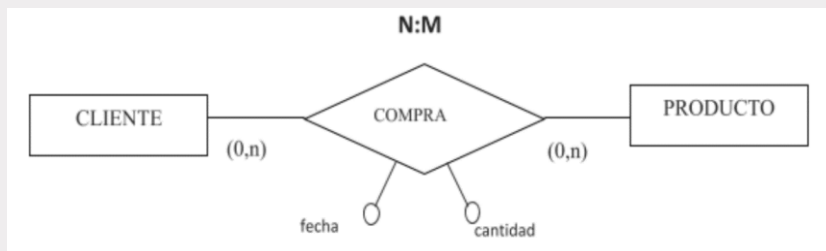




c) Atributos propios de una relación

También las relaciones pueden tener atributos.

Son aquellos atributos cuyo valor sólo se puede obtener en la relación, puesto que dependen de todas las entidades que participan en la relación.

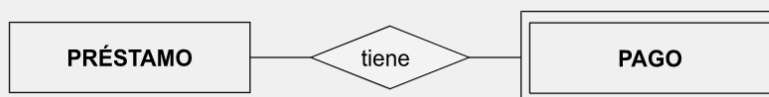


Entidades Fuertes y Entidades débiles

Una **entidad fuerte** es aquella que no necesita de otra entidad débil para existir.

Una **entidad débil** es aquella que sí que necesita de otra para existir.

Las entidades fuertes se representarán con un rectángulo y las débiles con un rectángulo de línea doble, por ejemplo:



PAGO			PRÉSTAMO	
Nro	Importe	Fecha	Nro	Total
1	\$1200	02/02/2020	1	\$100.000
2	\$2500	02/02/2020	2	\$125.000
1	\$1500	02/03/2020	3	\$230.000
1	\$1000	03/03/2929		



Supongamos el conjunto de entidades **Pago** que vemos en la imagen anterior, con los atributos *número*, *importe* y *fecha*. Se trata de almacenar la información de los distintos pagos necesarios para devolver un préstamo.

Cada pago es un pago distinto al resto de pagos. Pero en estas condiciones existen números de pagos repetidos, incluso el importe y la fecha podrían repetirse, como vemos en el ejemplo.

Los pagos son diferentes y corresponden a préstamos diferentes.

¿Cómo podemos identificar a qué préstamo corresponde un pago? Sencillamente no podemos, puesto que no hay una clave primaria que identifique cada pago. Nos encontramos ante un ejemplo de entidad débil, cada pago depende de la existencia de un préstamo para tener significado.

Una posible solución podría ser agregar una clave primaria a la entidad PAGOS que permita identificar cada pago, como se ve a continuación.

PAGO			
*ID	Nro	Importe	Fecha
23	1	\$1200	02/02/2020
24	2	\$2500	02/02/2020
25	1	\$1500	02/03/2020
26	1	\$1000	03/03/2929

En este caso si podríamos diferenciar cada pago del resto de pagos.

Sin embargo sigo sin poder darle significado, pues no puedo relacionar el pago con el préstamos. **Por ende Pagos es una entidad débil en existencia y en identificación.**

La razón es simple: existen pagos porque necesariamente hay un préstamo. ¿Porqué debería pagar sino me han prestado?.

Pero, entonces, ¿cómo podríamos dar significado a registros de una entidad débil y relacionarlos a una entidad fuerte?

PAGO			
Préstamo	Pago	Importe	Fecha
1	1	\$1200	02/02/2020
2	2	\$2500	02/02/2020
3	1	\$1500	02/03/2020
4	1	\$1000	03/03/2929
1	2	\$1200	02/03/2020
...

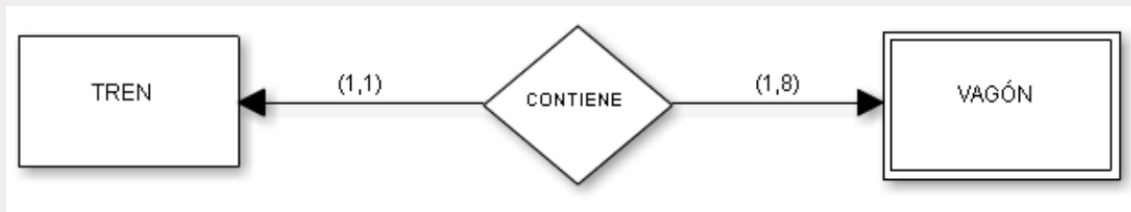
La clave primaria de la entidad débil se formará con la clave primaria de la entidad fuerte más el atributo que identifica a los datos de la entidad débil.



Relaciones de Dependencia

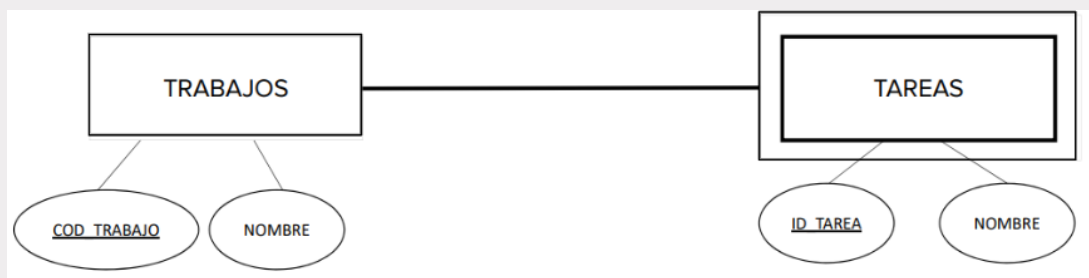
Las relaciones de dependencia son las que se producen entre las entidades fuertes y las débiles. Existen dos tipos, y algo ya te adelantamos con el ejemplo anterior:

- **De existencia:** La entidad débil no puede existir sin la fuerte de la que depende.

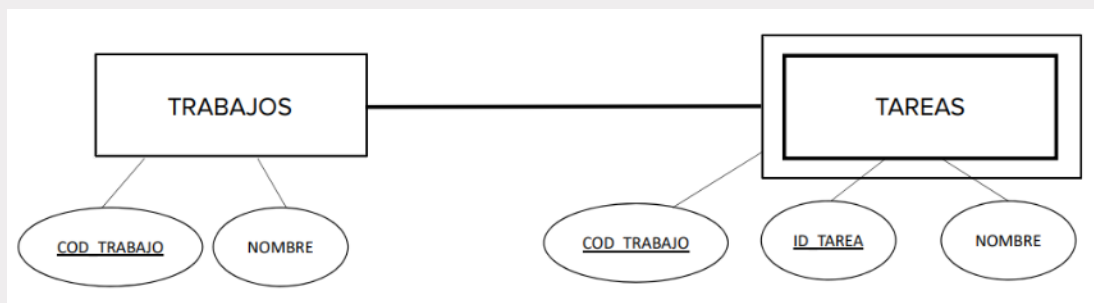


El vagón por sí solo puede trabajar, pero su existencia depende del tren.

- **De identificación:** Cuando además de por existencia, la entidad débil no es capaz de identificarse por sí misma y depende del identificador de la entidad fuerte.



Para identificar la tarea sumamos la clave primaria del Trabajo





La cardinalidad de una entidad débil siempre es la siguiente:

- **(1,1)**. En el lado de la entidad fuerte, haciendo referencia a que cada elemento de la entidad débil se relaciona con uno y solamente uno de los ejemplares de la entidad fuerte.
- **(0,n)** o **(1,n)**. En el lado de la entidad fuerte. Lo habitual es (0,n) indicando que los ejemplares de la entidad fuerte no tienen por qué relacionarse con ninguno de los ejemplares de la entidad débil.

Para comprender mejor la Cardinalidad, te dejamos algunos ejemplos prácticos (¿te animás a hacer los MER de estos ejemplos?):

➤ **Uno a Uno (1:1):**

Un empleado tiene un único número de seguro social asignado.

Un cliente tiene una única dirección de envío.

➤ **Uno a Muchos (1:N):**

Un autor puede escribir varios libros.

Un departamento tiene muchos empleados.

➤ **Muchos a Muchos (N:N):**

Varios estudiantes pueden inscribirse en varios cursos.

Varios productos pueden ser parte de varias órdenes de compra.

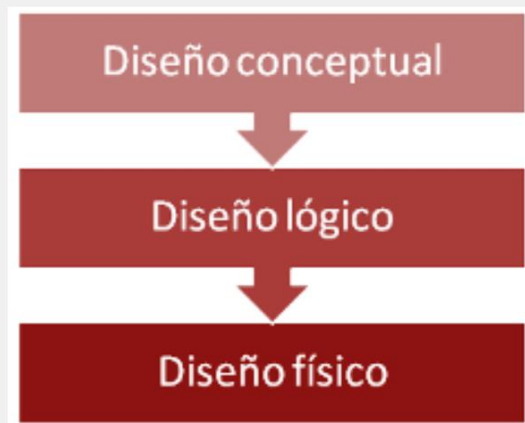
En los ejemplos anteriores, la cardinalidad se expresa en términos de relaciones entre entidades. Por ejemplo, en el caso del autor y el libro, un autor puede tener una relación de uno a muchos con los libros, lo que significa que un autor puede escribir varios libros (1:N). En el caso de los estudiantes y los cursos, hay una relación de muchos a muchos, lo que significa que varios estudiantes pueden inscribirse en varios cursos (N:N).

Es importante tener en cuenta que la cardinalidad puede variar según el contexto y los requisitos específicos del sistema de información o de la base de datos en particular. Es esencial determinar la cardinalidad correcta para establecer relaciones adecuadas entre las entidades y garantizar un modelo de datos preciso y eficiente.



Del modelo conceptual al Modelo Lógico

Luego de haber realizado el análisis de la situación real, modelando la misma mediante un Esquema Conceptual, a través del MER, nos abocaremos a la obtención del modelo lógico de nuestra Base de Datos.

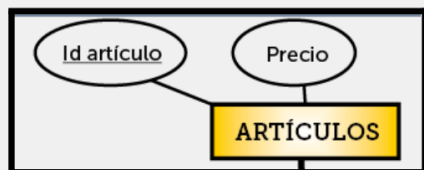


En este apunte te contamos los pasos a seguir detalladamente a partir de diversos ejemplos.

Entidades

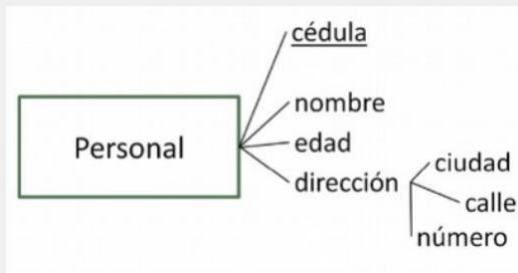
Todas las **entidades regulares** presentes en el MER se transforman en **tablas** en el modelo lógico, manteniendo el número y tipo de los atributos, así como las claves primarias.

- El **nombre** de la tabla será el nombre de la entidad.
- Cada **atributo** de la entidad será un **campo** de la tabla.
- Los atributos que definen la **clave primaria** serán, también, la **clave primaria** de la tabla.
- En el caso que hayan atributos compuestos se descompone el atributo en campos.
(lo vamos a ver mejor en el ejemplo de Personal)



ARTÍCULOS

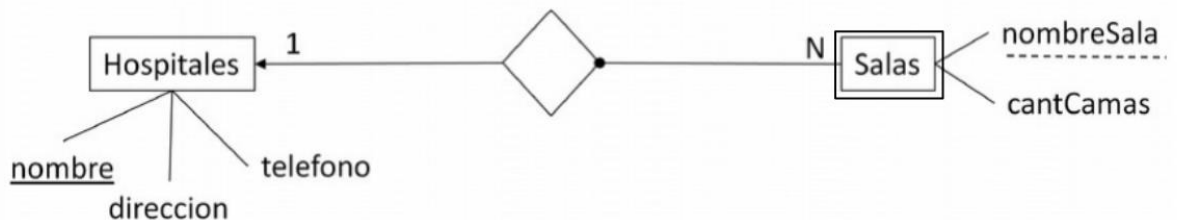
* ID_artículo
Precio



PERSONAL

*Cédula
Nombre
Edad
Ciudad
Calle
Número

Las **entidades débiles** también se convierten en tablas en el modelo relacional, manteniendo el **número y tipo de los atributos**, pero su clave primaria se forma por la composición de su clave primaria con la clave primaria de la entidad regular de la cual depende.



HOSPITAL

*NombreHospital
Dirección
Telefono

SALAS

*NombreHospital
*NombreSala
Dirección
Telefono



Relaciones

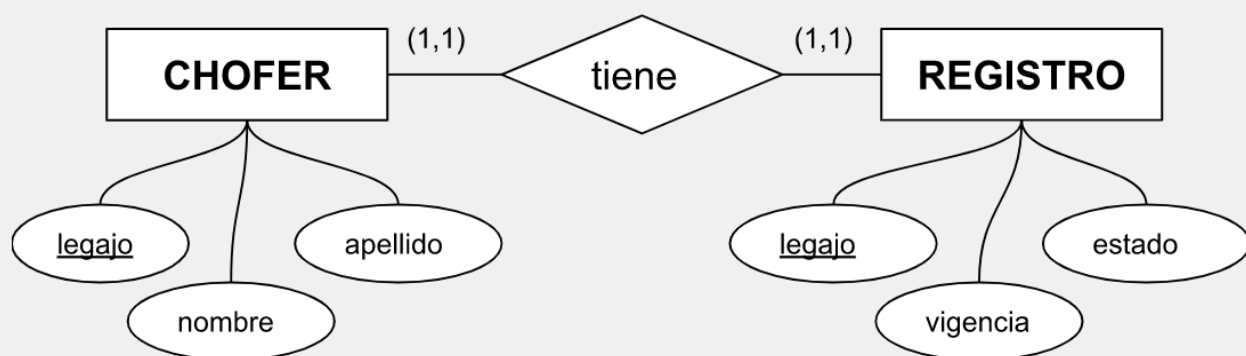
Para convertir las relaciones a tablas, hay que tener presente su cardinalidad: 1:1, 1-N, N:N

Dependiendo de cómo se definan las cardinalidades de las relaciones, éstas pueden generar una tabla o por el contrario traducirse en campos dentro de las tablas asociadas a las entidades que participan en dicha relación.

Relaciones 1:1

En este tipo de relaciones las entidades podrían utilizar la misma clave o cada cual la suya.

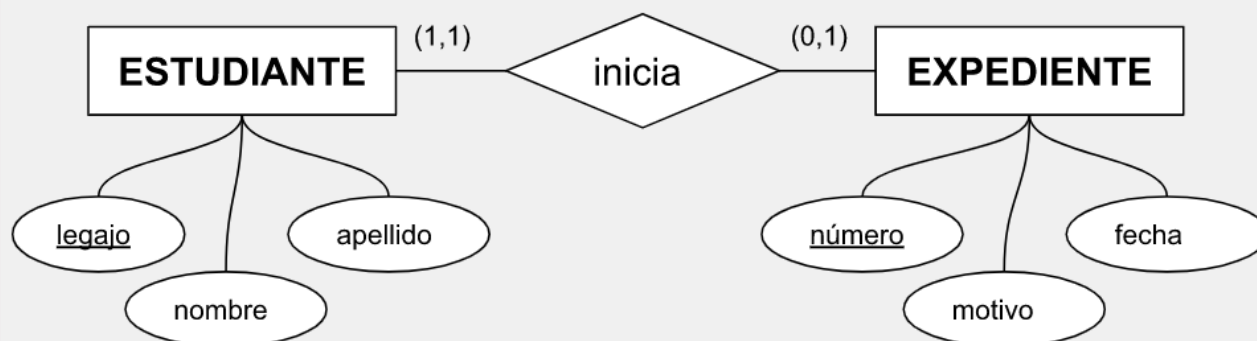
- Si las dos entidades tienen el mismo identificador, entonces se transforman en una única tabla por la agregación de los atributos de las dos entidades y la clave es la clave de las entidades (es la misma en ambas).



CHOFER
*legajo nombre apellido vigencia estado

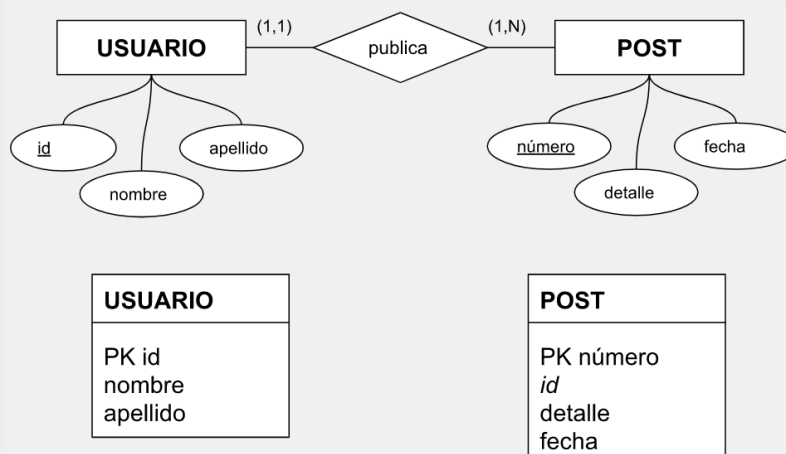


- Si las dos entidades tienen **distinto identificador**, entonces cada entidad se transforma en una tabla con clave principal y cada tabla tendrá como **clave foránea** el identificador de la otra tabla con la cual está relacionada.



Relaciones 1:N

En el tipo de relación binaria 1:N, **no es necesario generar otra tabla**, lo que se hace es incluir en la entidad del lado de N el identificador de la otra tabla y los posibles campos de la relación si los hubiere. Veamos cómo funciona:

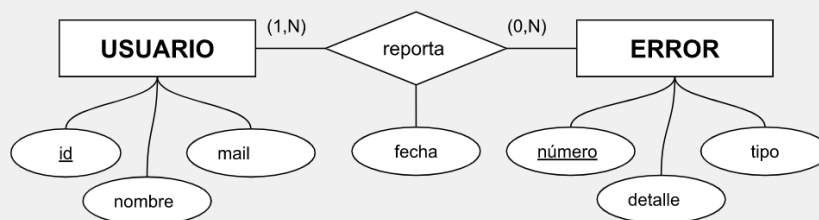




Relaciones N:N

En este caso crearemos una **nueva tabla** para representar la **relación**, agregando la clave primaria de cada una de las tablas, además de los campos que correspondan a atributos de la relación.

La **clave primaria** de la nueva tabla será la **combinación de las claves primarias** de las tablas que participan en la relación.



USUARIO	REPORTE	ERROR
PK id nombre mail	PK id PK número fecha	PK número detalle tipo

Relaciones de Herencia

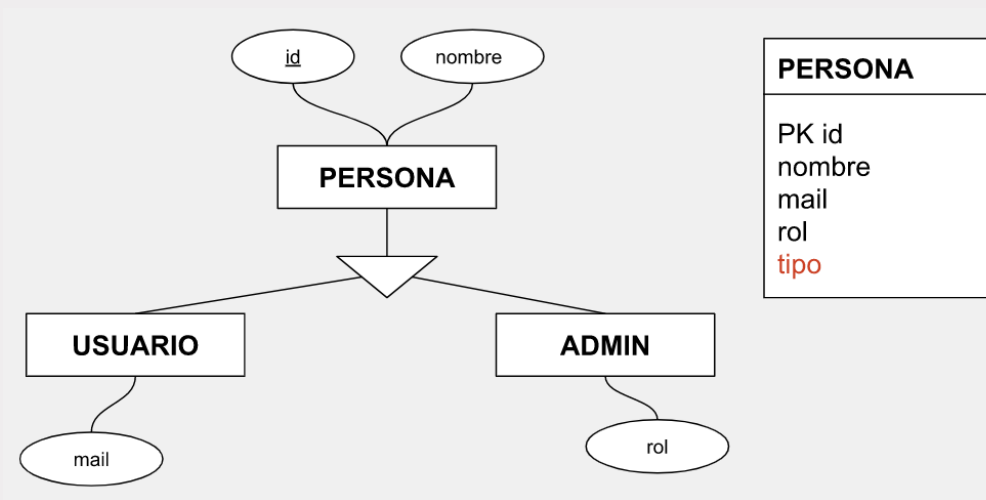
En el caso de representaciones de Herencia normalmente se generan las siguientes tablas: una para la super entidad (nivel superior), con los atributos comunes, y otra tabla para cada una de las entidades subordinadas, con la misma clave que la entidad superior y los atributos propios de esa sub-entidad. Por ejemplo

PERSONA	USUARIO	ADMIN
PK id nombre	FK id mail	FK id rol

Con esta solución un Usuario nunca puede ser Admin, ni viceversa....
qué opción hay para poder representar que es posible que un Usuario sea Admin?



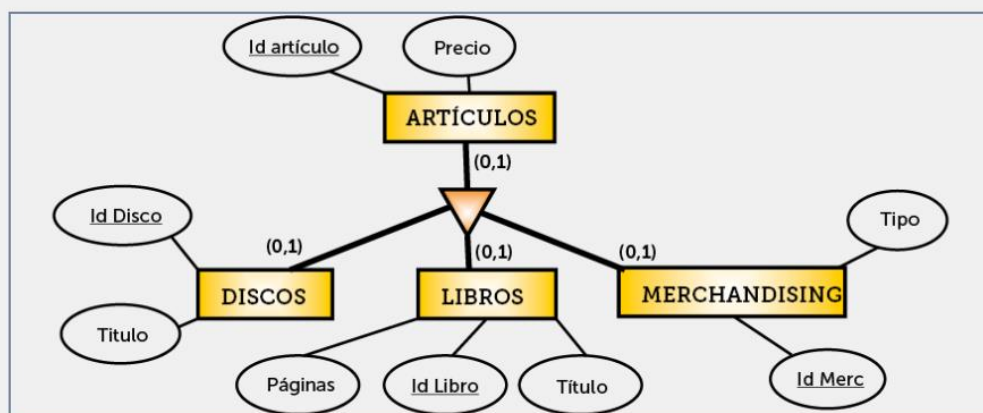
Para solucionarlo, tenemos la opción de crear una tabla con la super entidad, sus atributos y claves e incorporar los atributos de las entidades subordinadas. Agregando además el campo *tipo* que permitirá distinguir si la persona es USUARIO o ADMIN.



Especialización

La relación de Especialización implica la designación de subgrupos dentro de un conjunto de entidades.

Puede incluir subgrupos de entidades que se diferencian de alguna forma de las otras entidades del conjunto. Se produce un Diseño descendente: refinamiento a partir de un conjunto de entidades inicial en sucesivos niveles.



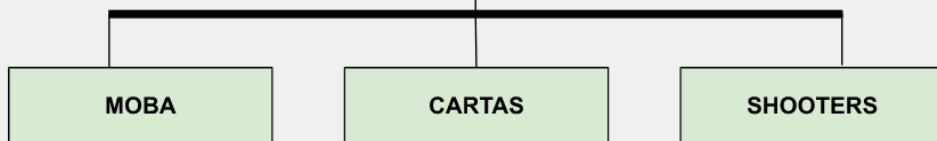
Por ejemplo, si debemos modelar la venta de artículos de una tienda.

Y los artículos pueden ser Discos, Libros o productos de Merchandising.

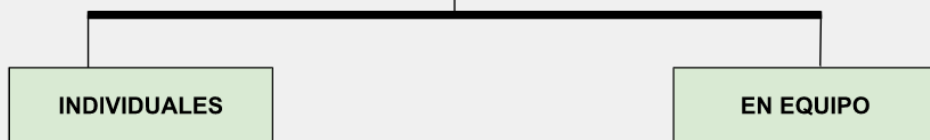
Se puede aplicar repetidamente la especialización para refinar el esquema de diseño. Por ejemplo:



DEPORTES ELECTRÓNICOS



DEPORTES ELECTRÓNICOS

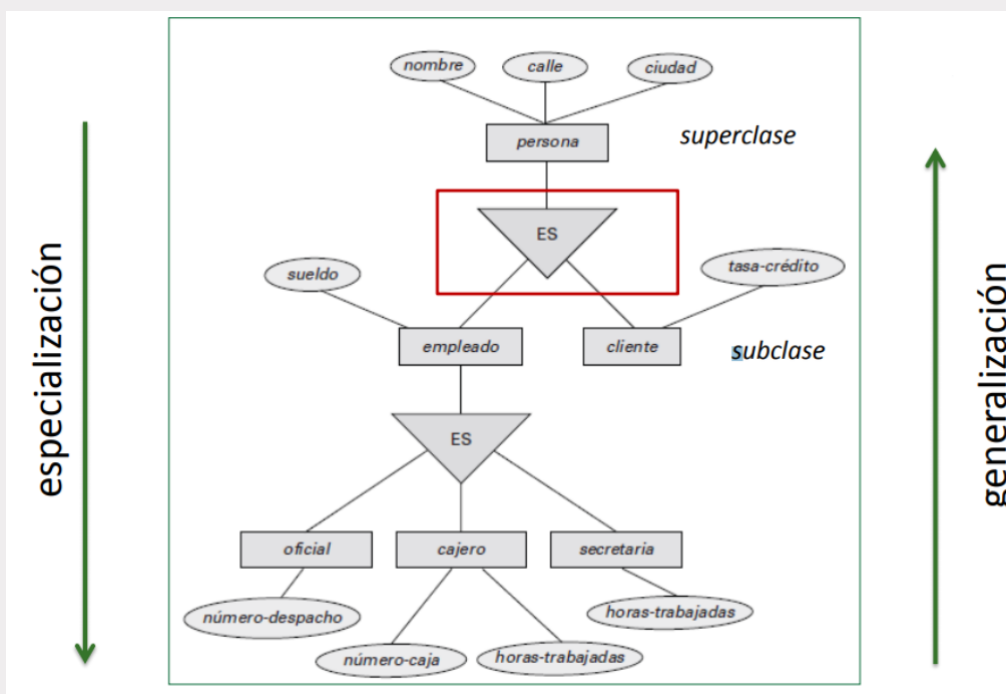


En el caso de los deportes electrónicos, uno puede ser del tipo Shooter y también puede ser del tipo En Equipo

Generalización

La generalización es una inversión simple de la especialización.

Gráficamente se produce un Diseño ascendente: varios conjuntos de entidades se sintetizan en un conjunto de entidades de nivel más alto basado en características comunes.





Herencia de atributos

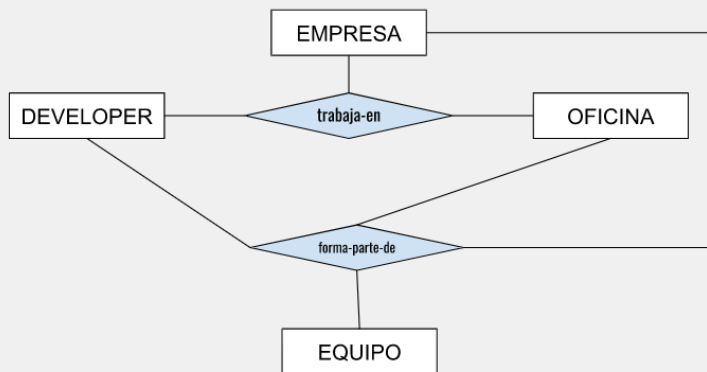
Los atributos de los conjuntos de entidades de nivel más alto son heredados por los conjuntos de entidades de nivel más bajo.

Por ejemplo: *cliente* y *empleado* heredan los atributos de la entidad *persona*.

Agregación

Una limitación del MER es que no resulta posible expresar relaciones entre relaciones.

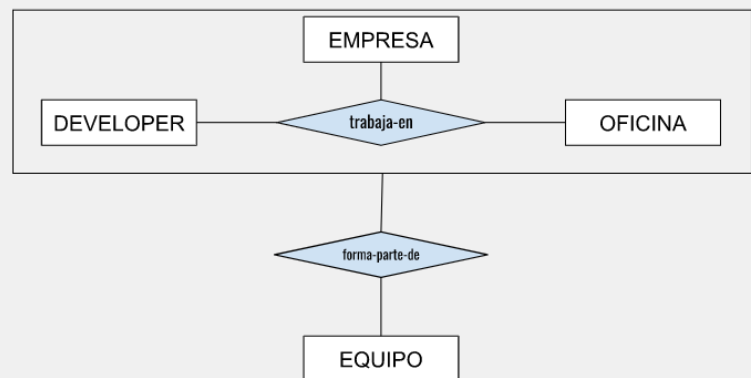
Para resolverlo se utiliza la Agregación, una abstracción a través de la cual las relaciones se tratan como entidades de nivel más alto.



Consideremos la relación ternaria ***trabaja-en***, entre empleado, sucursal y trabajo.

Vemos que se dificulta comprender las relaciones.

Usando la agregación, podríamos agrupar las entidades de la relación *trabaja-en* y obtenemos la siguiente representación





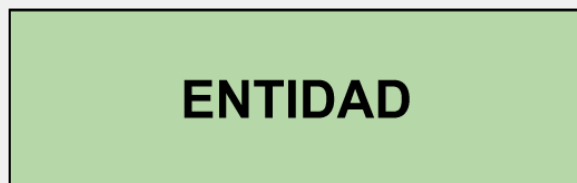
Notación de Chen

Peter Chen (también conocido como Peter Pin-Shan Chen) actualmente se desempeña como miembro de la facultad de la Universidad Carnegie Mellon ubicada en Pittsburgh y se le atribuye el desarrollo del modelo ER para el diseño de bases de datos en los 70. Mientras trabajaba como profesor adjunto en la Escuela de Administración y Dirección de Empresas Sloan del MIT, publicó un documento influyente en 1976 llamado "Modelo entidad-relación: hacia una visión unificada de los datos".

GLOSARIO DE SÍMBOLOS

Entidad

Algo que se puede definir, como una persona, objeto, concepto u evento, que puede tener datos almacenados acerca de este. Piensa en las entidades como si fueran sustantivos. Por ejemplo: un cliente, estudiante, auto o producto



Atributo

Una propiedad o característica de una entidad.



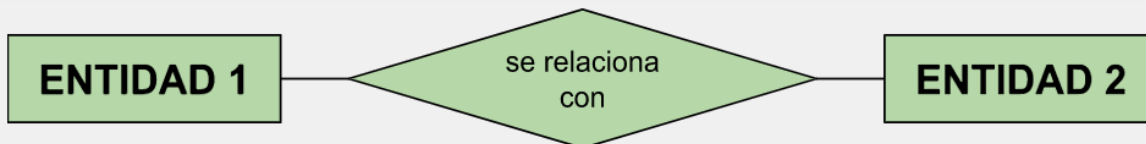
Categorías de los atributos: los atributos se clasifican en *simples*, *compuestos* y *derivados*, así como de *valor único* o de *valores múltiples*.

- **Simples:** significa que el valor del atributo es mínimo y ya no puede dividirse, como un número de teléfono.
- **Compuestos:** los subatributos surgen de un atributo.
- **Derivados:** los atributos se calculan o derivan de otro atributo, por ejemplo, la edad se calcula a partir de la fecha de nacimiento.
- **Valores múltiples:** se denota más de un valor del atributo, como varios números de teléfono para una persona
- **Valor único:** contienen solo un valor de atributo. Los tipos se pueden combinar, por ejemplo, puede haber atributos de valor único simples o atributos de múltiples valores compuestos.



Relaciones

Asociaciones que permiten vincular entidades entre sí. Usar verbos para relaciones y representarlas con rombos.



Cardinalidad

Define los **atributos numéricos** de la relación entre dos entidades o conjuntos de entidades. Las tres relaciones cardinales principales son **uno a uno**, **uno a muchos** y **muchos a muchos**.

Un ejemplo de uno a uno sería un estudiante asociado a una dirección de correo electrónico. Un ejemplo de uno a muchos (o muchos a uno, en función de la dirección de la relación) sería un estudiante que se inscribe en muchos cursos, y todos esos cursos se asocian a ese estudiante en particular. Un ejemplo de muchos a muchos sería los estudiantes en grupo están asociados a múltiples miembros de la facultad y a su vez los miembros de la facultad están asociados a múltiples estudiantes.

Vamos con un ejemplo para representar un **MER con Notación de Chen**:

Le contratan para hacer una BD que permita apoyar la gestión de un sistema de ventas. La empresa necesita llevar un control de proveedores, clientes, productos y ventas. Un proveedor tiene un RUT, nombre, dirección, teléfono y página web. Un cliente también tiene RUT, nombre, dirección, pero puede tener varios teléfonos de contacto. La dirección se entiende por calle, número, comuna y ciudad. Un producto tiene un id único, nombre, precio actual, stock y nombre del proveedor. Además, se organizan en categorías, y cada producto va sólo en una categoría. Una categoría tiene id, nombre y descripción. Por razones de contabilidad, se debe registrar la información de cada venta con un id, fecha, cliente, descuento y monto final. Además, se debe guardar el precio al momento de la venta, la cantidad vendida y el monto total por el producto.

