



**PROGRAMA EDUCATIVO
INFORMATICA ADMINISTRATIVA**

**UNIDAD DE APRENDIZAJE
ADMINISTRACIÓN DE BASES DE DATOS**

Unidad de competencia III
CONOCIMIENTO

METODOLOGÍAS PARA EL DISEÑO DE BASES DE DATOS

**ELABORACION
ADRIAN TRUEBA ESPINOSA**



PRESENTACIÓN DEL CURSO

La unidad de aprendizaje “Administración de Bases de Datos”, se imparte en el 6° semestre de la licenciatura en Informática Administrativa. Tiene la finalidad de desarrollar las competencias en los alumnos para la implementación de bases de datos, con del modelo de bases de datos relacional. Para ello, es necesario sentar las bases teóricas y metodológicas para el diseño de una base de datos relacional con el modelo entidad relación y relacional.



CONTENIDO DEL CURSO

Unidad 1: CONCEPTOS FUNDAMENTALES DE BASES DE DATOS

Unidad 2: MODELO DE DATOS

Unidad 3: METODOLOGÍA DE DISEÑO DE BASE DE DATOS

Unidad 4: CONCEPTOS DE DISEÑO DE APLICACIONES DE BASES DE DATOS

Unidad 5: DISEÑO E IMPLEMENTACIÓN DE BASES DE DATOS

Unidad 6: ADMINISTRACIÓN Y SEGURIDAD EN BASE DE DATOS

Unidad 7. TENDENCIAS ACTUALES EN BASES DE DATOS



METAS A ALCANZAR

Que el alumno desarrolle las competencias técnicas y de especialidad para el análisis, diseño e implementación de bases de datos

Modelo Relacional.



OBJETIVO DEL MATERIAL DIDÁCTICO

Que el alumno conozca el método de la para modelar bases de datos para la implementación física en un Sistema Gestor de Bases de Datos



METODOLOGÍA DEL CURSO

El curso se desarrollará bajo el siguiente proceso de estudio:

1. Exposición de parte del profesor mediante la utilización de este material en diapositivas.
2. Diseñar con el Modelo entidad relación el modelo conceptual
3. Diseñar con el Modelo lógico relación del el modelo Relacional
4. Implementación física de la base de datos en un SGBD
5. Participación en clases
6. Prácticas de laboratorio



UTILIZACIÓN DEL MATERIAL DE DIAPOSITIVAS

El material didáctico visual es una herramienta de estudio que sirve como una guía para que el alumno repase los temas más significativos de “la metodología de diseño de bases de datos”, cabe aclarar que será un tutor el cual proporcionará las ideas generales del tema, asiendo ejercicios en el salón de clase.



Unidad de competencia III

CONOCIMIENTO METODOLOGÍAS PARA EL DISEÑO DE BASES DE DATOS



Procedimiento general

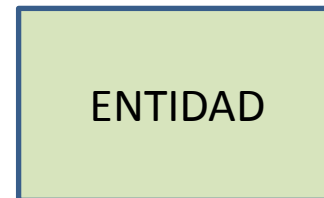
Para el diseño de la base de datos se iniciara en primera instancia con el **Modelo entidad relación**, después se abordara el **modelo relacional** para plantear el **modelo lógico** y se implementara la **base de datos física** en un SGBD



El **modelo entidad-relación** ER es un **modelo** de datos que **permite representar cualquier abstracción, percepción y conocimiento en un sistema de información** formado por un conjunto de objetos denominados **entidades** y relaciones, incorporando una representación visual conocida como **diagrama entidad-relación**.

Se representa por un rectángulo

Entidad. Las entidades constituyen las tablas de la base de datos que permiten el almacenamiento de las tuplas.





Atributos: Son las características, rasgos y propiedades de una entidad, que toman un valor particular.

Los atributos de una tabla son el predicado que permite definir lo que decimos de un determinado sujeto.

Por ejemplo de una entidad o tabla catálogo, se pueden determinar los atributos *título*, *subtítulo*, *título paralelo*, *otras formas del título*, *autor principal*, *otras menciones de responsabilidad*, *edición*, *mención de edición*, *editorial*, *lugar de publicación*, *fecha de publicación*,...

Se representa por un ovalo



Si el atributo es una llave se subraya



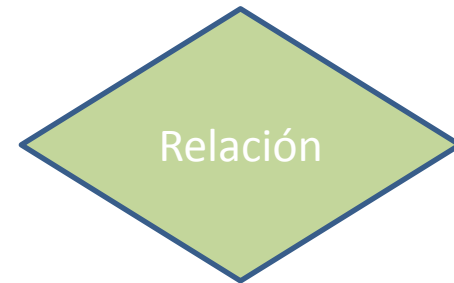


Relación. Vínculo que permite definir una dependencia entre los conjuntos de dos o más entidades.

Es la relación entre la información contenida en los registros de varias tablas.

Por ejemplo, los usuarios suelen clasificarse según una lista de tipos de usuarios, ya sean profesores, alumnos o investigadores. De esta forma es posible emitir la relación entre el usuario *Jorge Martínez* como *alumno* y *Enrique Valtierra* como *profesor*.

La relación se representa mediante un rombo





Interrelación. Las interrelaciones las constituyen los vínculos entre entidades (tablas), de forma tal que representan las relaciones definidas en el esquema relacional de forma efectiva, a través de un campo clave que actúa como código de identificación y referencia para relacionar.

Interrelación = cardinalidad

La Interrelación se representa con una flecha o un línea



flecha interrelación **1**

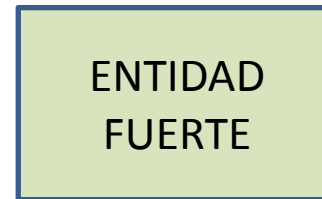
línea interrelación **N**



Entidades fuertes. Lo constituyen las tablas principales de la base de datos que contienen las tuplas principales y requieren de entidades auxiliares para completar su descripción o información.

Por ejemplo la entidad usuario es una entidad fuerte en relación a la tabla tipos de usuarios, que es una entidad débil dada su condición auxiliar para clasificar a los usuarios registrados en la biblioteca.

Se representa por un rectángulo

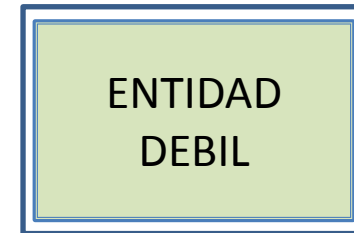




Entidades débiles. Son entidades débiles las tablas auxiliares de una tabla principal a la que complementan la información de un registros relacionado.

Por ejemplo también son consideradas entidades débiles las tablas intermedias que sirven para compartir información de varias tablas principales.

Se representa por un rectángulo doble





•**Clave.** Es el atributo de una entidad que tiene como objetivo distinguir cada registro del conjunto, sirviendo sus valores como datos vinculantes de una relación entre entidades.

Clave principal primaria. Permiten identificar unívocamente cada registro de una entidad. Por ejemplo campo auto-numérico interno ID.

•**Clave externa o foranea.** Campo clave conformado por el valor de una clave principal primaria de otra tabla. Por ejemplo el campo *id_tipodeusuario* en la tabla *usuarios* es un campo clave externo que guarda el valor del campo primario ID de la tabla *tipodeusuario*, especificando de esa forma que un usuario como *Enrique Valtierra* sea de tipo 2 es decir *profesor*.



Integridad referencial. Se denomina integridad referencial al tipo de interrelación que se produce entre tablas mediante un atributo clave que deberá contener la cadena alfanumérica exacta al identificador de la tabla auxiliar para poder realizar la relación entre los registros.

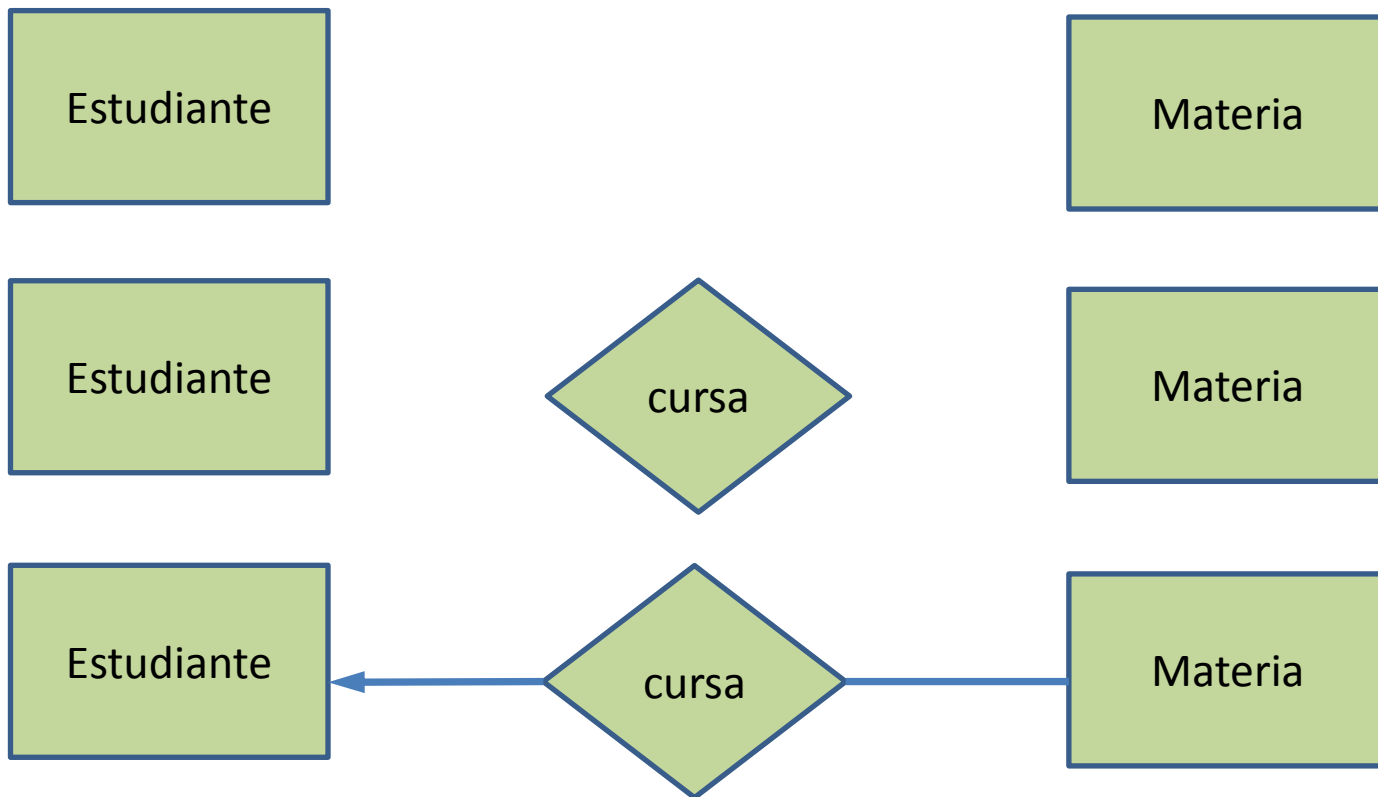
Si se borra la clave principal de una entidad fuerte y no se borra en todas las otras entidades donde esta como clave primaria se rompe la integridad referencial.

Para evitar este problema se debe borrar en todas las entidades y se conoce como borrar en cascada.



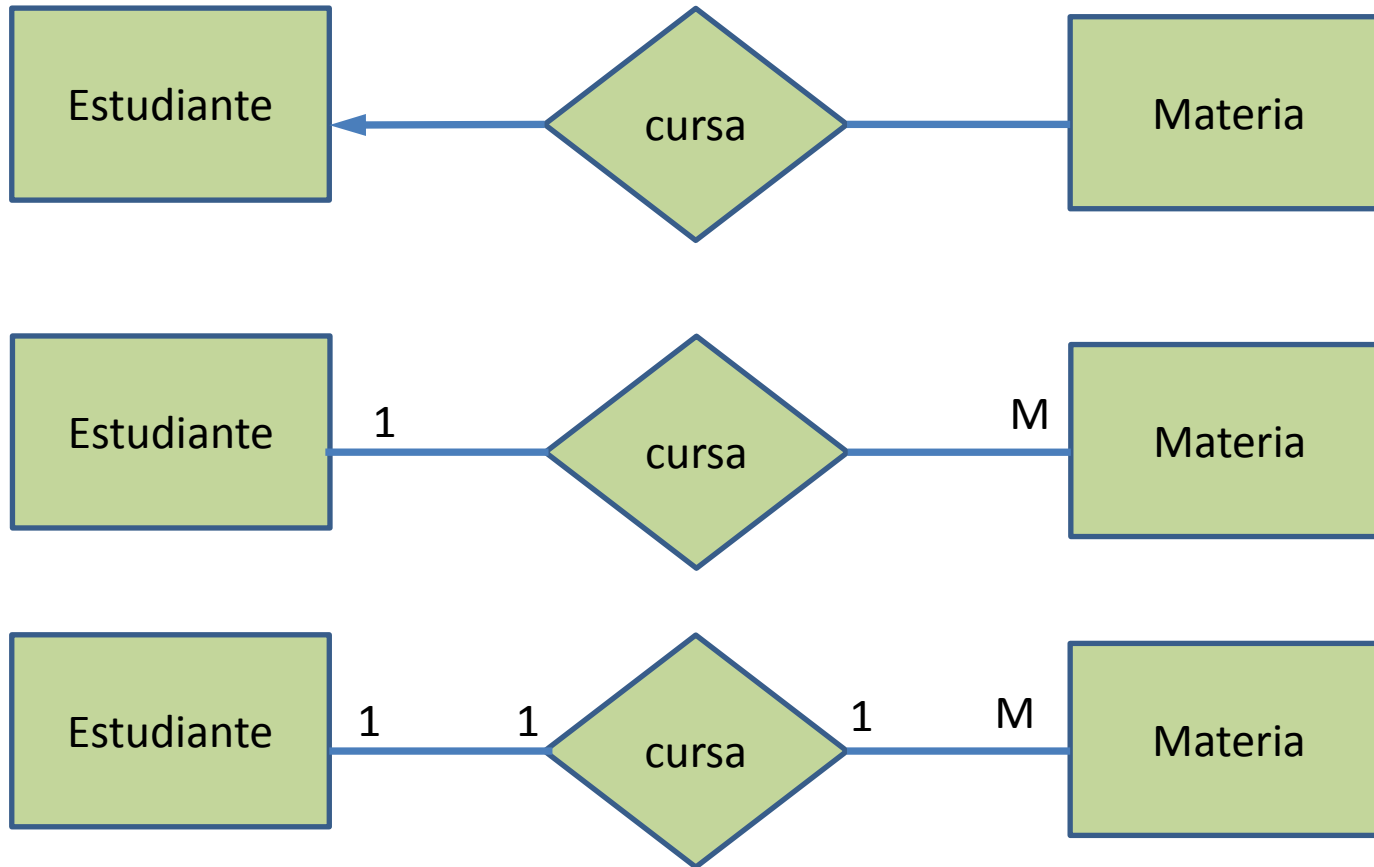
Considerando los elementos expuesto considere lo siguiente:

Deseamos modelar la relación que hay entre un estudiante y la materias que cursa





En la literatura también lo pueden encontrar como

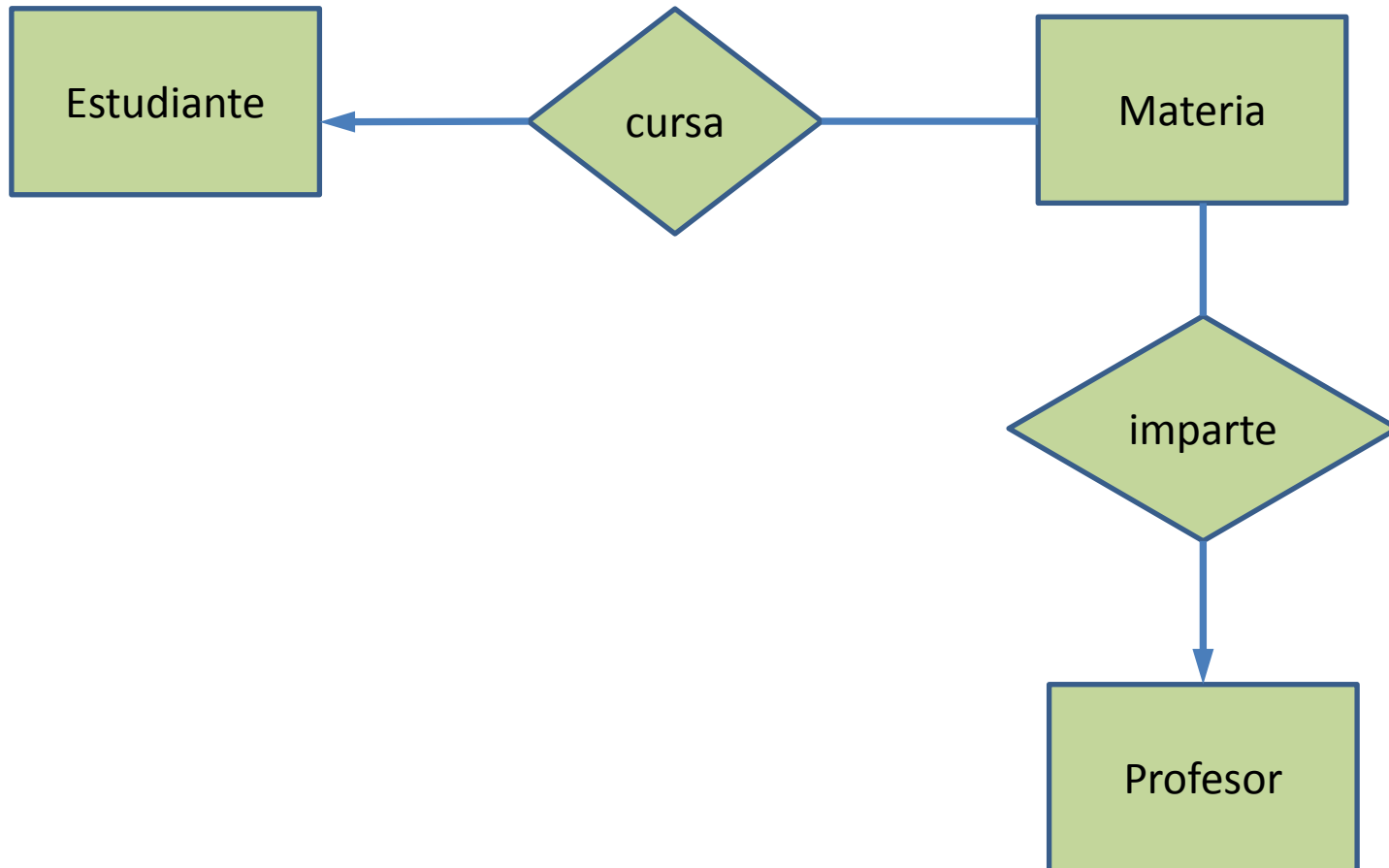


La M
puede ser
substituid
a por una
N o un *

En este curso se usara la primera notación

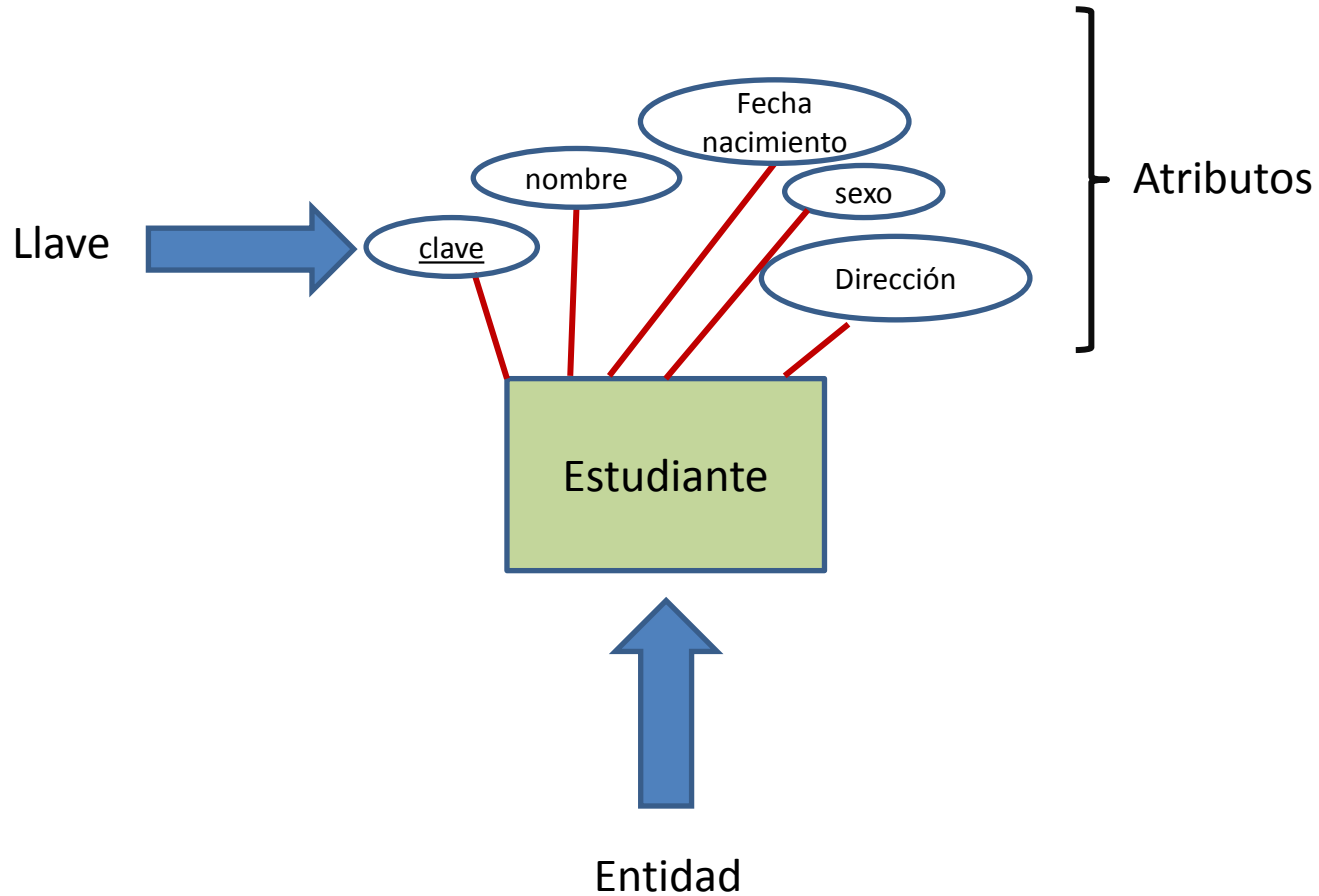


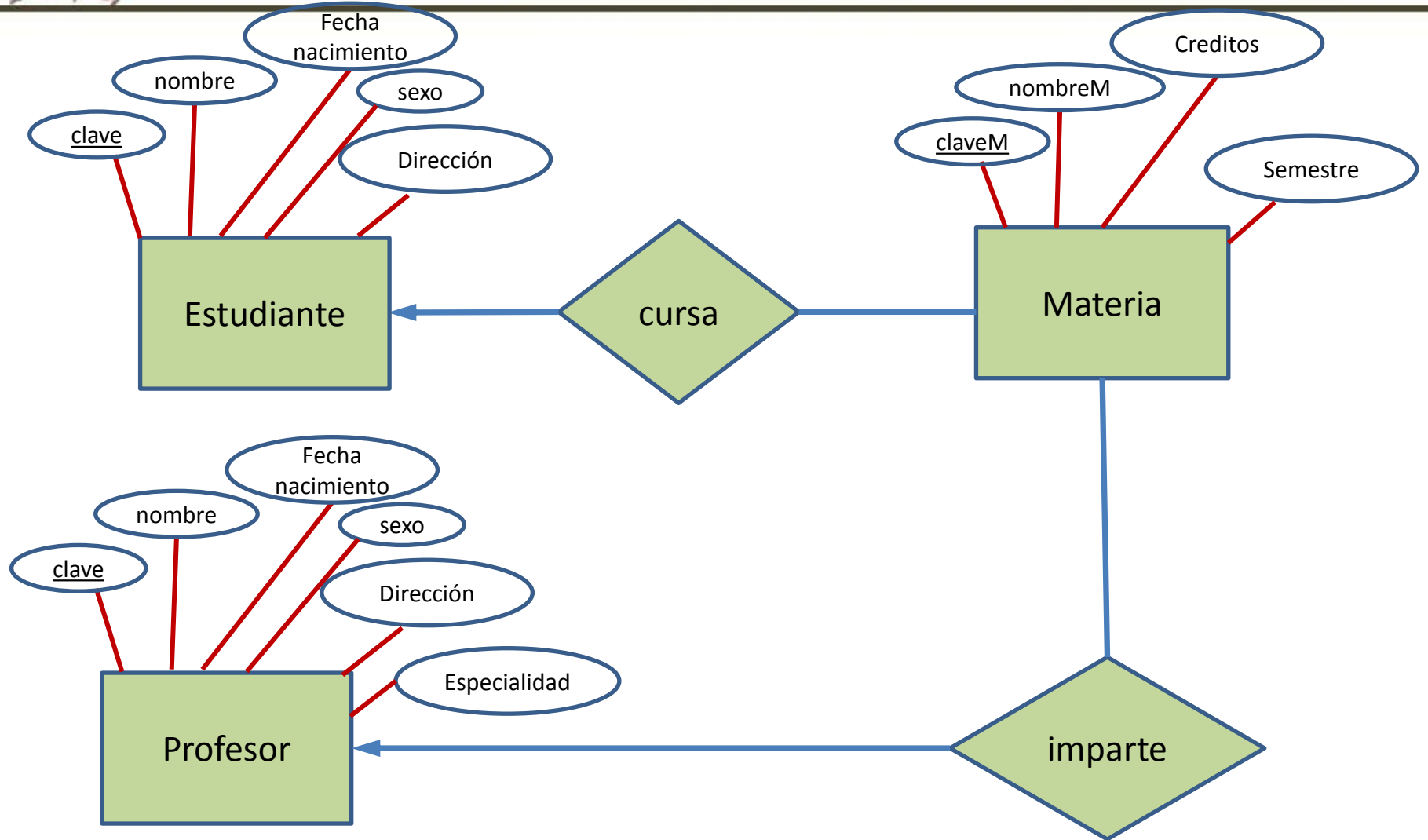
Para modela la relación que hay entre un estudiante, la materias que cursa y el profesor que la impartió entonces, quedaría de la siguiente manera





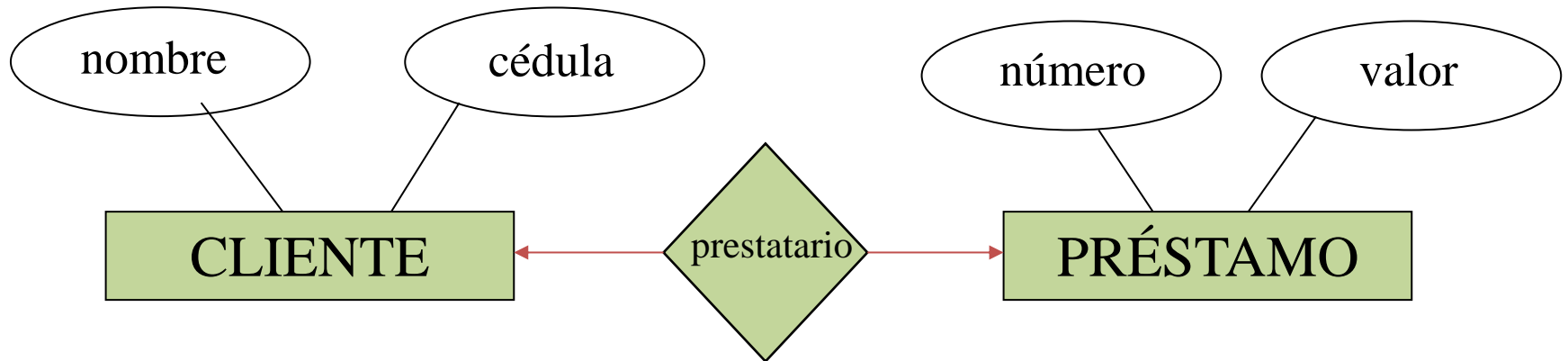
Si se consideran los atributos para una entidad quedaría de la siguiente manera







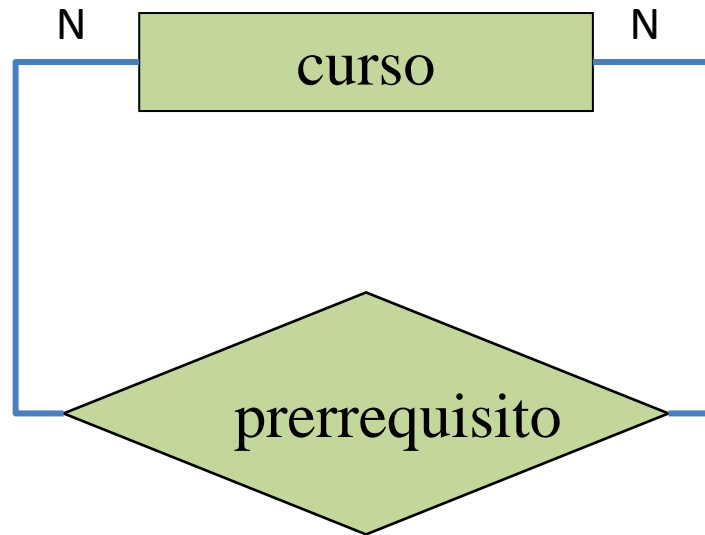
Otro ejemplo podría ser:





Auto relación

- Relaciones entre una entidad y sí misma



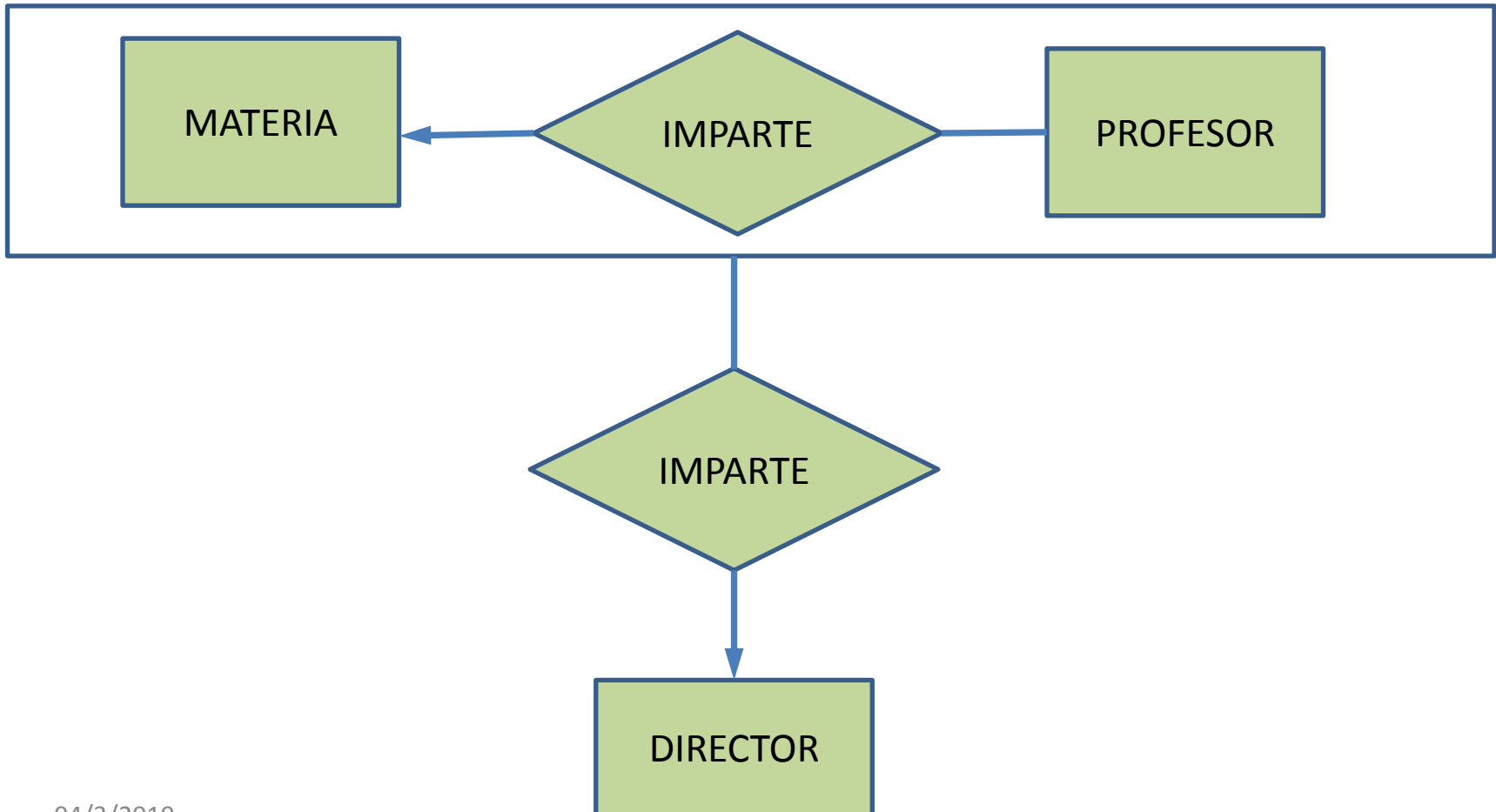


Agregación

- Al suponer que hay un DIRECTOR que SUPERVISA la impartición de clases de cada MATERIA por parte del los PROFESOR
- Se diera que: Entidades DIRECTOR, DOCENTE y MATERIA. Una relación DOCENTE-IMPARTE-MATERIA
- ¿Cómo modelamos la supervisión?
- Lo que el director supervisa, no es el profesor ni la materia, sino la impartición de la materia por parte del profesor.



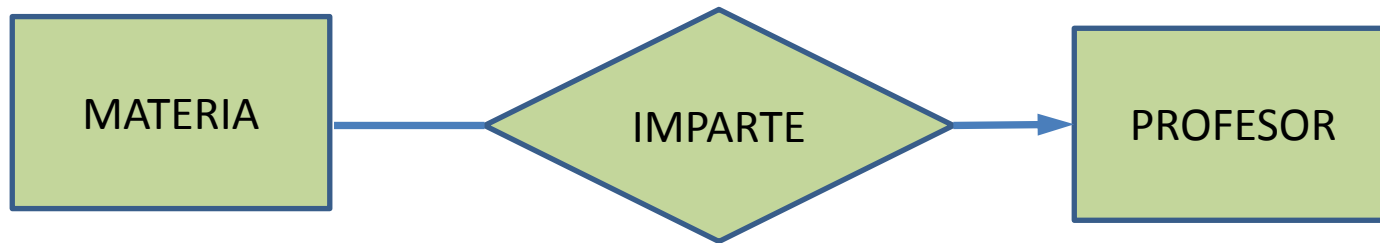
Para este tipo de problemas, se puede considerar la relación como si fuera una entidad. A esto lo llamamos agregación





Totalidad

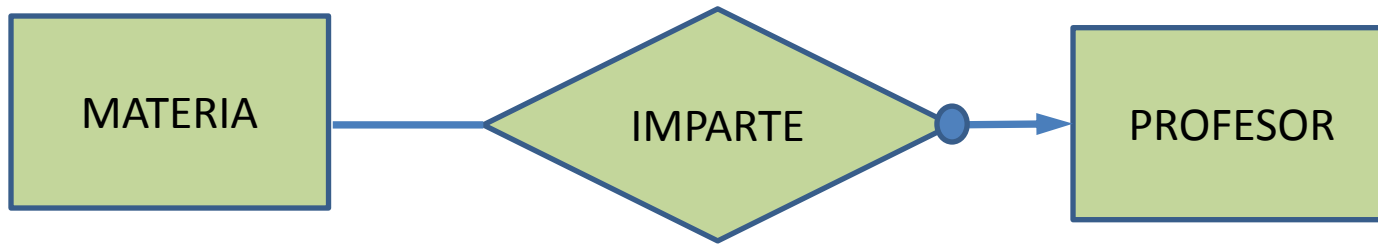
- Imagine que no queremos mantener PROFESORES a menos que IMPARTAN una MATERIA



- Lo que queremos es una forma de decir que todo PROFESOR debe IMPARTIR al menos una MATERIA (o que una entidad de PROFESOR sólo puede existir si participa en una instancia de la relación IMPARTE).
- N significa de 0 a N, queremos decir de 1 a N o “al menos 1”



Se representa esto con un punto en el rombo de la relación, del lado de la entidad que tiene participación total



- Note que en una relación en general, no todas las instancias de una entidad deben participar. En un caso de participación total como en el ejemplo, todas las instancias de PROFESOR deben participar en la relación PROFESOR-IMPARTE -CURSO (o de otra forma, deben IMPARTIR al menos una MATERIA).



Atributos

- Otro aspecto importante de los atributos, es si son determinantes. Diremos que un atributo es determinante cuando no pueden existir en el conjunto de entidades, dos entidades que tengan el mismo valor para ese atributo.
- Por ejemplo, la cédula puede ser considerada determinante. Representaremos los atributos determinantes subrayándolos.





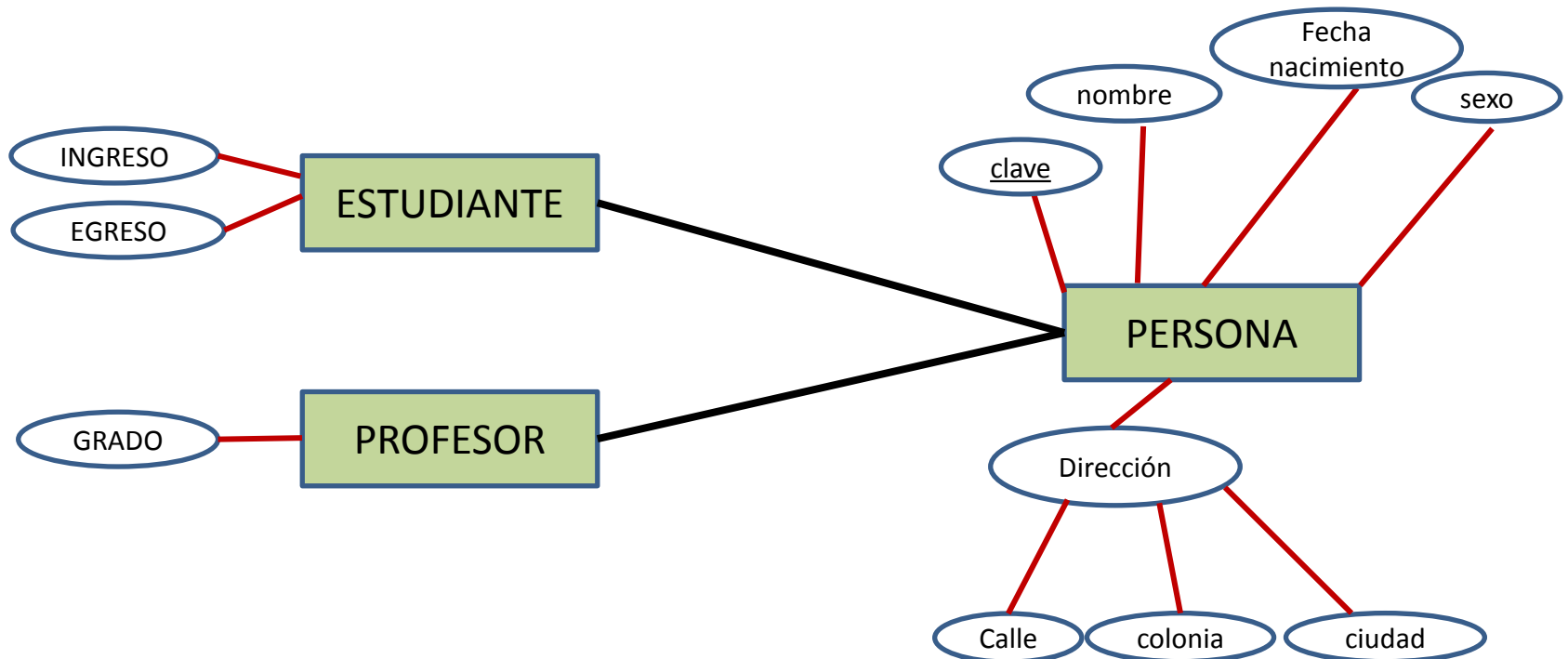
Generalización / Especialización

- La especialización permite modelar sub-entidades.
- Por ejemplo, donde modelamos entidades ESTUDIANTE y PROFESORES. En ambos casos, podríamos tener atributos en común como cédula, nombre, apellido y muchos otros.
- Esto es porque las dos entidades se pueden ver como especializaciones de una entidad más general: PERSONA. Las sub-entidades pueden tener sus propios atributos y participar en relaciones.



Generalización / Especialización (2)

- Las sub-entidades heredan los atributos de la entidad más general





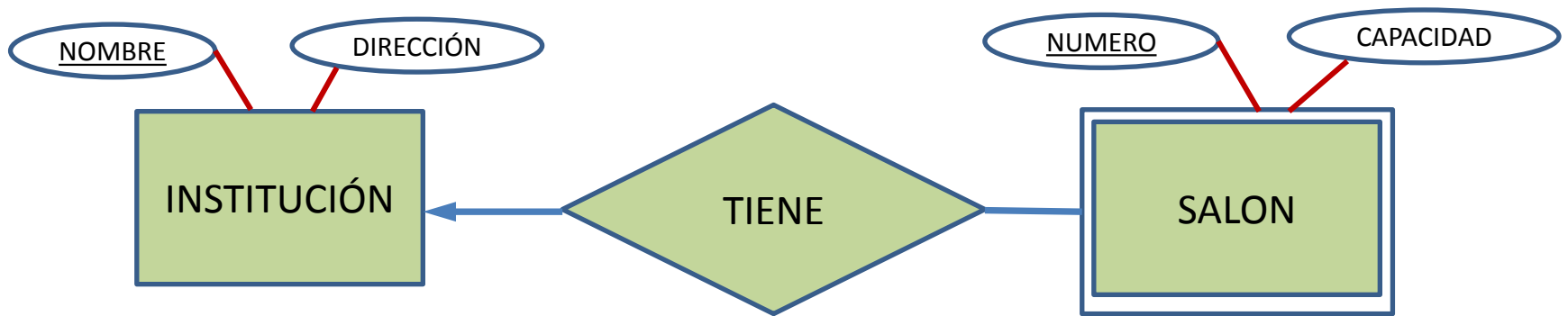
Entidad

- Una entidad puede no tener por sí misma datos suficientes como para poder identificarla
- Por ejemplo, los salones de un centro educativo se pueden identificar por número de salón, e.g. “salón 501”, pero los números podrían repetirse fuera del centro
- El salón queda identificado cuando además del número de salón, se tiene el nombre del centro, e.g. “salón 501 de la Facultad de Ingeniería”



Entidad débil

- En este caso diremos que la entidad SALON es débil respecto de INSTITUCIÓN, y lo representaremos con una flecha hacia la entidad INSTITUCIÓN (entidad fuerte).



- Las entidades débiles surgen naturalmente en cualquier relación de pertenencia: salones de un centro educativo, ciudades de un departamento



Restricciones no estructurales

- Muchas restricciones se pueden incluir en el lenguaje gráfico del Modelo Entidad-Relación, pero muchas otras no.
- Podríamos querer agregar al modelo restricciones del tipo: “la fecha de egreso de un estudiante, si está especificada, debe ser mayor que la fecha de ingreso”.
- Las escribiremos al pie del MER, en español

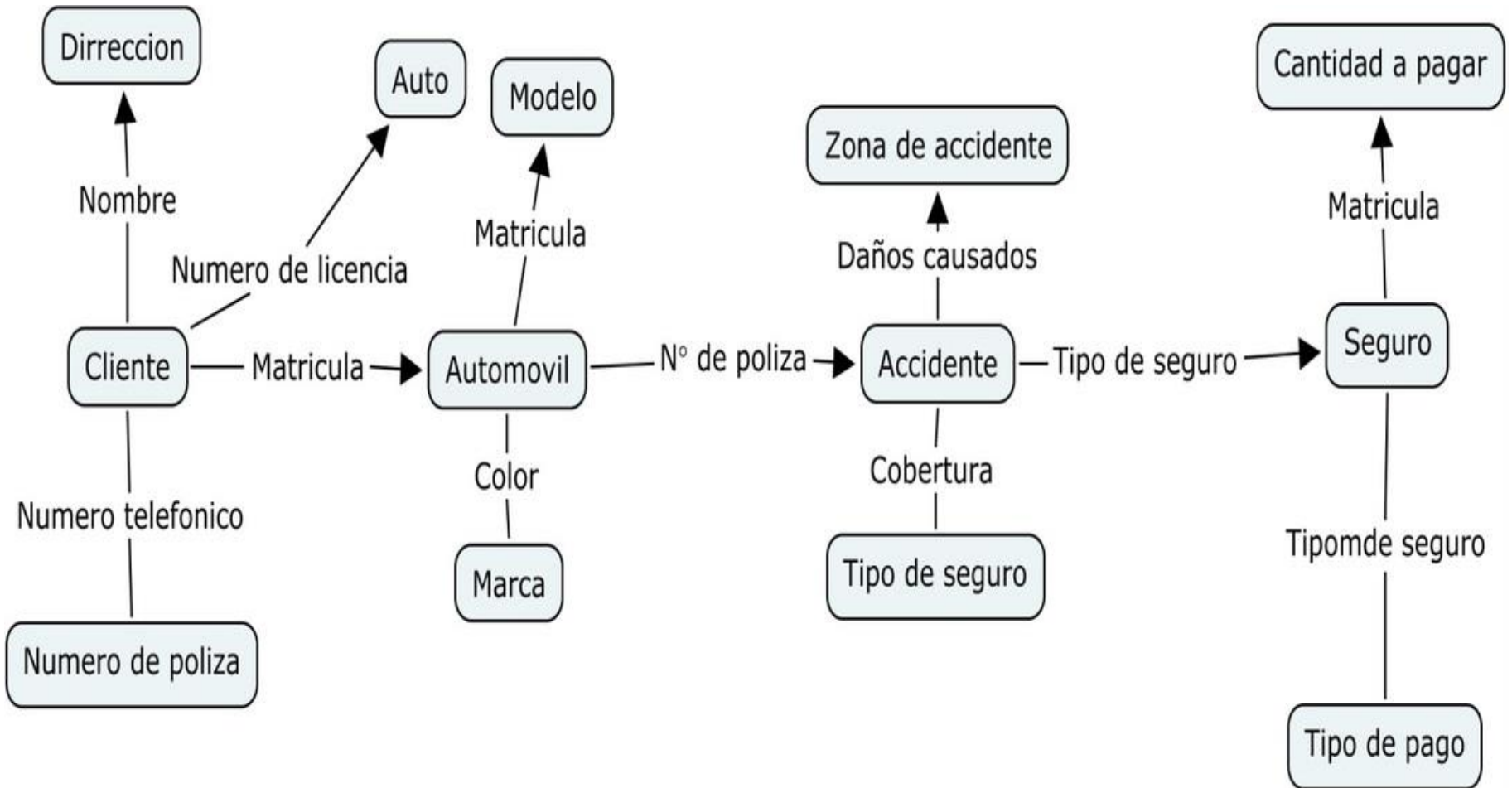


Restricciones no estructurales

- Siempre hay que pensar sobre una posible RNE en el caso de generalización: ¿las sub-entidades son una partición de la entidad más general?
- Siempre hay que pensar sobre una posible RNE en el caso de ciclos: ¿el ciclo debe cerrarse? Por ejemplo: “los empleados reservan habitaciones del hotel donde trabajan”

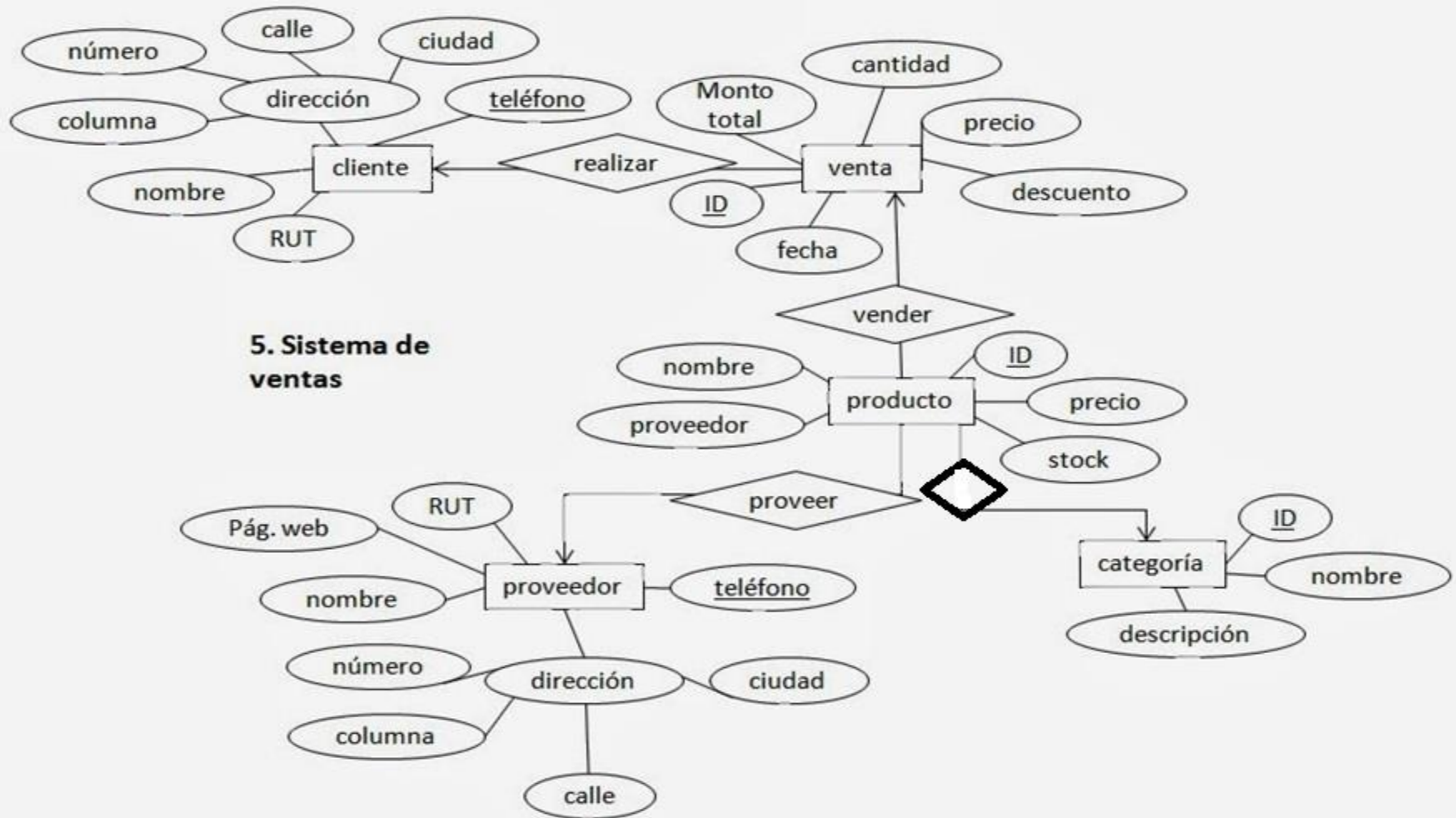


Diagrama seguro de autos





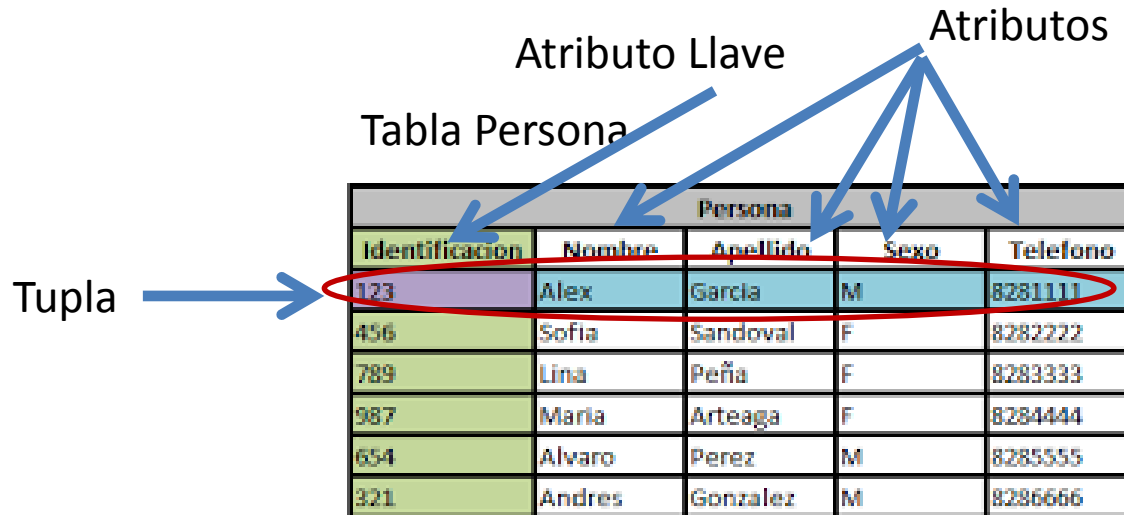
5. Sistema de ventas





Para un seguimiento se consideran el modelo anterior con el que se implemente la base de datos relación

La nomenclatura de las bases de datos es





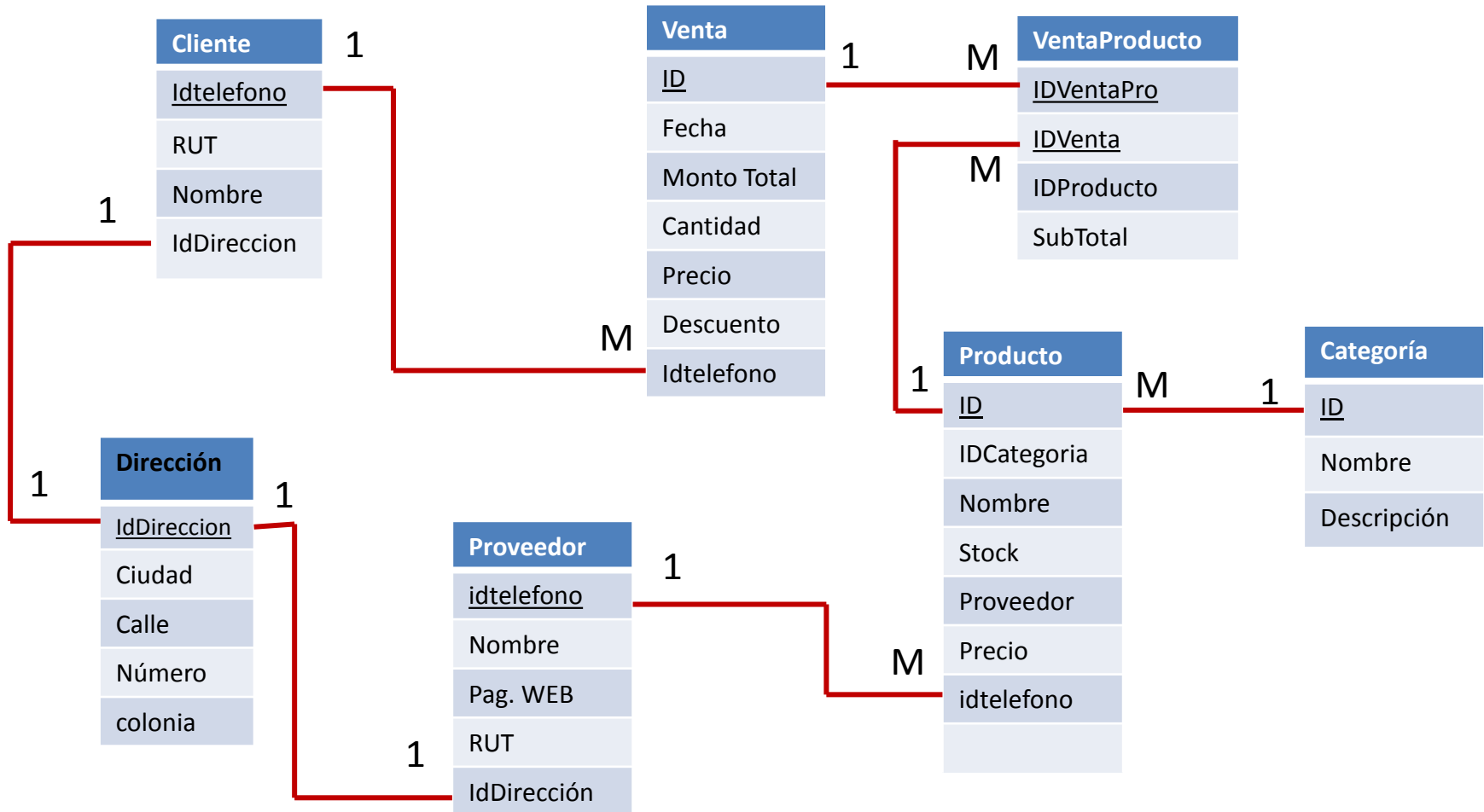
Tomamos la entidad cliente y la pasamos al modelo relacional



Con la consideración de que un cliente solo tiene una dirección



Generalizando todo el modelo quedaría de la siguiente manera





Nótese que aumento una tabla esto debido a que no se debe intrigar una llave foránea a cualquiera de las dos tablas ya que se haría redundancia



También observe que no hay un subtotal de la compra de cada producto en la venta total

Se podrían agregar tablas llave en cada una de Las relaciones y la base de datos funcionaria, sin embargo seria una base menos eficiente ya que Se tendría que entrar a mas tablas



Implementación física

Para la implementación física se tendría que ejecutar el siguiente código en el SGBD MySQL

```
DROP TABLE IF EXISTS `Categoria`;  
CREATE TABLE `Categoria` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `NombreA` varchar(20) DEFAULT NULL,  
  `Descripción` varchar(20) DEFAULT NULL,  
  PRIMARY KEY (`id`));
```

Categoría
<u>ID</u>
Nombre
Descripción



```
DROP TABLE IF EXISTS `Direccion`;  
CREATE TABLE `Direccion` (  
  `idDireccion` int(11) NOT NULL AUTO_INCREMENT,  
  `Ciudad` varchar(20) DEFAULT NULL,  
  `Calle` varchar(20) DEFAULT NULL,  
  `Colonia` varchar(20) DEFAULT NULL,  
  `numero` int(11) DEFAULT NULL,  
  PRIMARY KEY (`idDireccion`));
```

Dirección
<u>IdDireccion</u>
Ciudad
Calle
Número
colonia



```
DROP TABLE IF EXISTS `Cliente`;  
CREATE TABLE `Cliente` (  
  `idtelefono` int(11) NOT NULL,  
  `RUT` int(11) NOT NULL,  
  `Nombre` varchar(25) NOT NULL,  
  idDireccion int(11),  
  PRIMARY KEY `idtelefono`  
  KEY `fk_idDireccion` (`idDireccion`), CONSTRAINT  
  `fk_idDireccion` FOREIGN KEY (`idDireccion`) REFERENCES  
  `Direccion` (`idDireccion`) ON DELETE CASCADE ON UPDATE  
  CASCADE CASCADE ON UPDATE CASCADE);
```

Cliente
<u>Idtelefono</u>
RUT
Nombre
IdDireccion



```
DROP TABLE IF EXISTS `Proveedor`;  
CREATE TABLE `Proveedor` (  
  `idtelefono` int(11) NOT NULL,  
  `Nombre` varchar(25) NOT NULL,  
  `Pag. WEB` int(11) NOT NULL,  
  `RUT` int(11) NOT NULL,  
  idDireccion int(11),  
  PRIMARY KEY `idtelefono`  
  KEY `fk_idDireccion` (`idDireccion`), CONSTRAINT  
  `fk_idDireccion` FOREIGN KEY (`idDireccion`) REFERENCES  
  `Direccion` (`idDireccion`) ON DELETE CASCADE ON UPDATE  
  CASCADE CASCADE ON UPDATE CASCADE);
```

Proveedor
<u>idtelefono</u>
Nombre
Pag. WEB
RUT
IdDirección



```
DROP TABLE IF EXISTS `Venta`;  
CREATE TABLE `Venta` (  
  `ID` int(11) NOT NULL,  
  `Fecha` date NOT NULL,  
  `Monto` Decimal(11,2) NOT NULL,  
  `Cantidad` int(11) NOT NULL,  
  `Precio` Decimal(11,2) NOT NULL,  
  `Descuento` int(11) NOT NULL,  
  idtelefono int(11),  
  PRIMARY KEY `ID`,  
  KEY `fk_idtelefono`(`idtelefono`), CONSTRAINT  
  `fk_idDireccion` FOREIGN KEY (`idDireccion`) REFERENCES  
  `Direccion` (`idDireccion`) ON DELETE CASCADE ON UPDATE  
  CASCADE CASCADE ON UPDATE CASCADE);
```

Venta
<u>ID</u>
Fecha
Monto Total
Cantidad
Precio
Descuento
Idtelefono



```
DROP TABLE IF EXISTS `Producto`;
CREATE TABLE `Producto` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `IDCategoria` int(11) NOT NULL,
  `Nombre` varchar(30) NOT NULL,
  `Stock` int(11) DEFAULT NULL,
  `Proveedor` Varchar(30) DEFAULT NULL,
  `Precio` Decimal(10,2) NOT NULL,
  `Idtelefono` int(11) NOT NULL,
  PRIMARY KEY (`id`,`IDCategoria`,`idtelefono`),
  KEY `fk_IDCategoria` (`ID`),
  KEY `fk_Idtelefono` (`Idtelefono`),
  CONSTRAINT `fk_IDCategoria` FOREIGN KEY (`IDCategoria`)
REFERENCES `Categoria` (`idAlumno`) ON DELETE CASCADE ON
UPDATE CASCADE,
  CONSTRAINT `fk_idtelefono` FOREIGN KEY (`idtelefono`)
REFERENCES `Proveedor` (`idtelefono`) ON DELETE CASCADE ON
UPDATE CASCADE);
```

Producto
<u>ID</u>
IDCategoria
Nombre
Stock
Proveedor
Precio
idtelefono



VentaProducto

IDVentaPro

IDVenta

IDProducto

SubTotal

```
DROP TABLE IF EXISTS `VentaProducto`;  
CREATE TABLE `Producto` (  
  `idVentaPro` int(11) NOT NULL AUTO_INCREMENT,  
  `idVenta` int(11) NOT NULL,  
  `IDProducto` int(11) NOT NULL,  
  `SubTotal` Decimal(10,2) NOT NULL,  
PRIMARY KEY (`idVentaPro`, `idVenta`, `idProduct`),  
KEY `fk_IdVenta` (`IDVenta`),  
KEY `fk_IdProducto` (`IdProducto`),  
CONSTRAINT `fk_IDVenta` FOREIGN KEY (`IDVenta`)  
REFERENCES `Venta` (`idVenta`) ON DELETE CASCADE ON UPDATE  
CASCADE,  
CONSTRAINT `fk_IDProducto` FOREIGN KEY (`IDProducto`)  
REFERENCES `Producto` (`ID`) ON DELETE CASCADE ON UPDATE  
CASCADE);
```



Resumen

La metodología usada para el análisis y diseño de una base de datos usado en este ejercicio, es usado por personas que no tienen mucha experiencia en la creación de base de datos.

Se puede hacer el planteamiento considerando la construcción de modelo relacional sin pasar por el modelo entidad relación

Este método puede conllevar a pasar por alto algunas redundancias dentro de la tabla que podrían ser mejoradas, si después hacemos la normalización



LECTURAS RECOMENDADAS

INTRODUCCION A LAS BASES DE DATOS RELACIONALES [MARIA ANTONIA NEVADO CABELLO](#) , VISION NET, 2016. ISBN 9788498868098

BASES DE DATOS RELACIONALES Y MODELADO DE DATOS. [JOSE MANUEL PIÑEIRO GOMEZ](#) , S.A. EDICIONES PARANINFO, 2017. ISBN 9788428333566

**BASES DE DATOS RELACIONALES: FUNDAMENTOS Y DISEÑO LOGICO
VV.AA. , UNIVERSIDAD PONTIFICIA COMILLAS, 2015. ISBN 9788484681724**



UAEM

Universidad Autónoma
Del Estado de México

Centro Universitario
UAEM Texcoco



Gracias por su atención
contacto
atruebae@hotmail.com