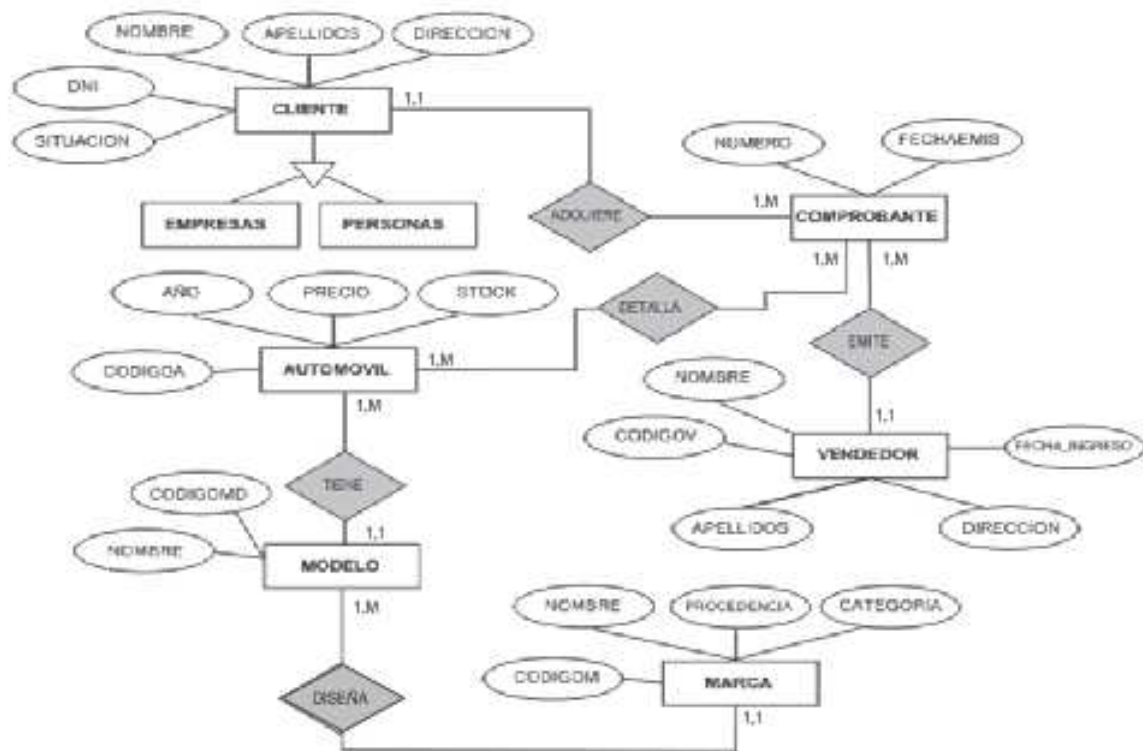


REALIZAMOS EL DIAGRAMA ENTIDAD RELACION



Esto es el Modelo Conceptual representado en un Diagrama Entidad Relación, lo que sigue ahora es realizar el Modelo Lógico de datos representado por un Modelo Entidad Relación.

El MER a un nivel lógico, comprende otros elementos asociados a las entidades y sus interrelaciones:

- ✓ Atributos
- ✓ Claves o Llaves
- ✓ Tipos de relaciones
- ✓ Cardinalidad

## ➤ MODELO LOGICO

### ✓ MODELO ENTIDAD RELACION (MER)

Los modelos lógicos basados en objetos se usan para describir datos en el nivel conceptual y el externo. Se caracterizan porque proporcionan capacidad de estructuración bastante flexible y permiten especificar restricciones de datos. Los modelos más conocidos son el modelo entidad-relación y el orientado a objetos.

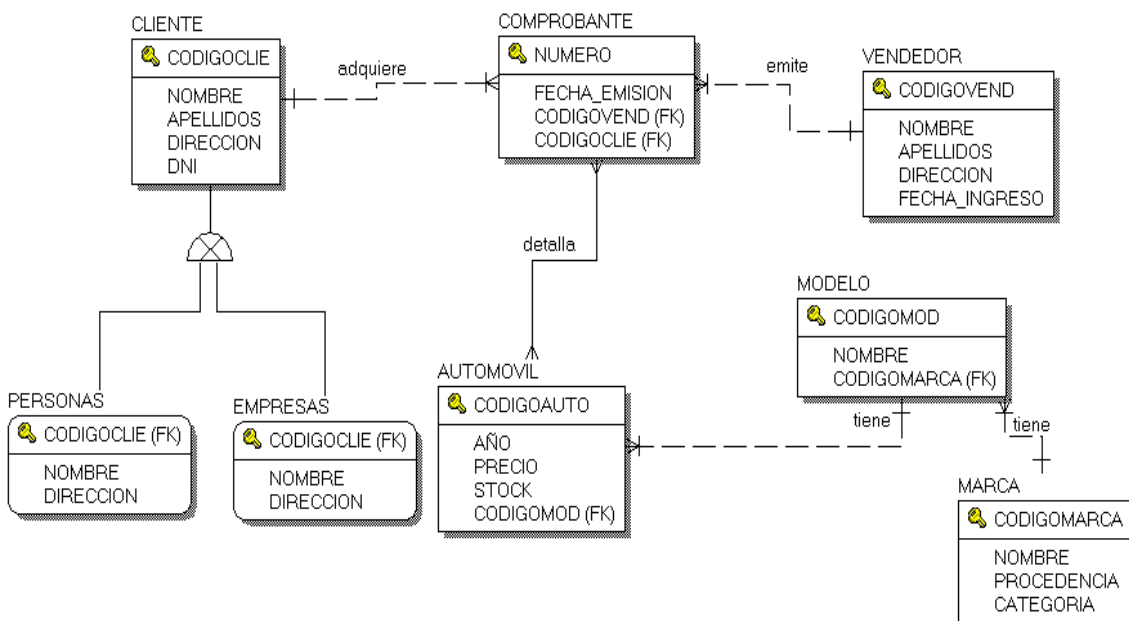
Actualmente, el más utilizado es el modelo entidad-relación, aunque el modelo orientado a objetos incluye muchos conceptos del anterior, y poco a poco está ganando mercado. La mayoría de las BBDD relacionales añaden extensiones para poder ser relacionales-orientadas a objetos.

### CREACIÓN DEL MODELO LÓGICO DEL CASO ANTERIOR

Lo primero que debemos hacer es identificar los atributos de las entidades, colocándoles los atributos que serán las claves primarias.

- ✓ Cliente: Códigoclie, nombre apellidos, dirección, DNI, situación.
- ✓ Vendedor: Codigovend, nombre, apellidos, dirección, fecha\_ingreso.
- ✓ Automóvil: Códigoauto, año, precio, stock.
- ✓ Comprobante: numero, fecha\_emisión.
- ✓ Modelo: Codigomod, descripción.
- ✓ Marca: Codigomarca, nombre, procedencia, categoría.

El Modelo Entidad Relación...



En este modelo todos los datos son almacenados en relaciones, y como cada relación es un conjunto de datos, el orden en el que estos se almacenen no tiene relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar por un usuario no experto. La información puede ser recuperada o almacenada por medio de consultas que ofrecen una amplia flexibilidad y poder para administrar la información.

Este modelo considera la base de datos como una colección de relaciones. De manera simple, una relación representa una tabla que no es más que un conjunto de filas, cada fila es un conjunto de campos y cada campo representa un valor que interpretado describe el mundo real. Cada fila también se puede denominar tupla o registro y a cada columna también se le puede llamar campo o atributo.

Para manipular la información utilizamos un lenguaje relacional, actualmente se cuenta con dos lenguajes formales el Álgebra relacional y el Cálculo relacional. El Álgebra relacional permite describir la forma de realizar una consulta, en cambio, el Cálculo relacional sólo indica lo que se desea devolver.

El lenguaje más común para construir las consultas a bases de datos relacionales es SQL, Structured Query Language o Lenguaje Estructurado de Consultas, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales.

El Modelo Lógico está muy orientado a registros, proporciona una vista más cercana a la estructura de la base de datos, que vendría a ser el modelo Físico de datos.

## ✓ **METODOLOGÍA DE DISEÑO DE BASES DE DATOS**

El diseño de una base de datos es un proceso complejo que abarca decisiones a muy distintos niveles. La complejidad se controla mejor si se descompone el problema en subproblemas y se resuelve cada uno de estos subproblemas independientemente, utilizando técnicas específicas. Así, el diseño de una base de datos se descompone en diseño conceptual, diseño lógico y diseño físico.

El diseño conceptual parte de las especificaciones de requisitos de usuario y su resultado es el esquema conceptual de la base de datos. Un esquema conceptual es una descripción de alto nivel de la estructura de la base de datos, independientemente del SGBD que se vaya a utilizar para manipularla. Un modelo conceptual es un lenguaje que se utiliza para describir esquemas conceptuales. El objetivo del diseño conceptual es describir el contenido de información de la base de datos y no las estructuras de almacenamiento que se necesitarán para manejar esta información.

El diseño lógico parte del esquema conceptual y da como resultado un esquema lógico. 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 SGBD. Un modelo lógico es un lenguaje usado para especificar esquemas lógicos (modelo relacional, modelo de red, etc.). El diseño lógico depende del tipo de SGBD que se vaya a utilizar, no depende del producto concreto.

## ➤ **MODELO FÍSICO**

El diseño físico parte del esquema lógico y da como resultado un esquema físico. 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 SGBD concreto y el esquema físico se expresa mediante su lenguaje de definición de datos.

En el modelo físico de datos, las entidades son consideradas tablas, los atributos vienen a ser los campos de las tablas, los elementos como registros o datos de las tablas, además agregamos el término de dominio de datos, es aquí donde se resuelve la relación

## ✓ **DOMINIOS**

Se refiere al conjunto de valores posibles que puede tener un atributo o campo o grupo de atributos o campos de una entidad o tabla. Cada atributo está asignado a uno de cuatro dominios básicos o primitivos, que son:

- Caracter o texto - String.
- Date o Fecha - Datetime
- Numero o Integer - Number
- Lógico - Blob

Los dominios primitivos son la base para formar otros dominios más complejos definidos por el usuario, dependerá también del gestor de base de datos (plataforma) sobre la cual se apoyará la data.

✓ **EXTENSION**

Indica el número máximo de caracteres o dígitos para cada uno de sus valores, podemos considerar que esto va a ser un subconjunto del dominio de un atributo, dado que el número de caracteres o dígitos restringe el conjunto posible de valores para el atributo o campo.

✓ **VALORES PERMITIDOS**

El conjunto de valores permitidos para un atributo describe exhaustivamente los valores potenciales de un atributo, es decir, un conjunto de valores posibles que el atributo soportará como data. Por ejemplo:

Unidad\_venta= TM (tonelada métrica), RO (rollo), BO (bolsa), PQ (paquete)

✓ **VALOR A ALGORITMO POR OMISION**

Para cada atributo que pueda contener valores permitidos se puede especificar un algoritmo por omisión o bien un valor por omisión (pero no ambos). Por ejemplo:

Codigo\_cliente = Codigo\_cliente + 1

✓ **ALGORITMO POR DERIVACION**

Solamente podemos especificar algoritmos de derivación para atributos derivados. En la práctica el diseñador debe tomar la decisión sobre si un atributo derivado debe ser calculado o almacenado por memoria. Por ejemplo:

Total\_ventaitem = valor\_venta + igv

Total\_venta = total\_ventaitem

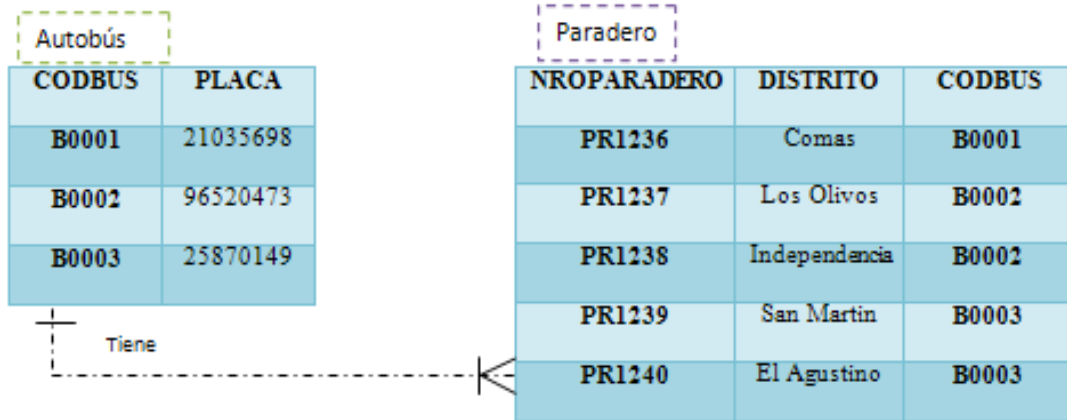
Una entidad en el modelo físico tiene esta forma:

CODIGO_CLIE	NOMBRE	DIRECCION	TELEFONO
C0001	AUGUSTO	SAN JUAN	2567348
C0002	JORGE LUIS	LINCE	9453889

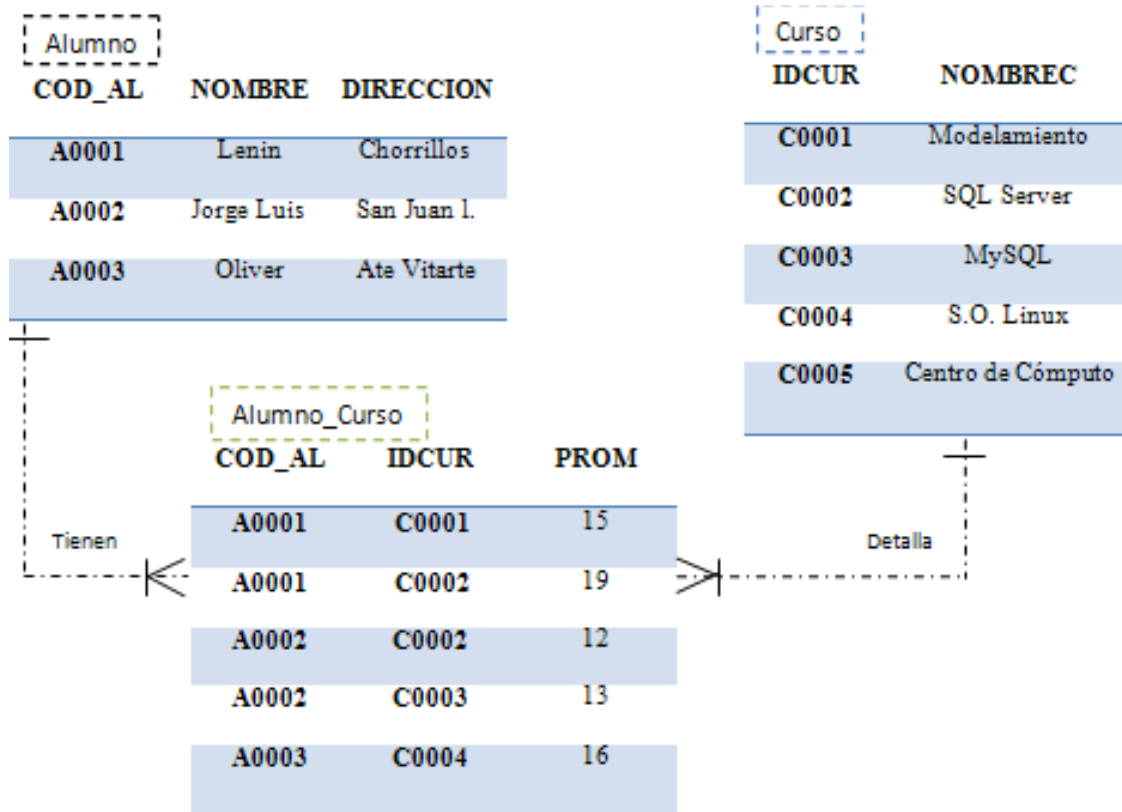
Esta entidad nos representa una tabla de nuestra base de datos, con los dominios para cada atributo o columna (campo), por defecto aparecen todos como Char (caracter), para asignar un tipo de dato (dominio) a cada uno se utiliza las opciones de la herramienta case Erwin, más adelante veremos cómo hacerlo. La tabla con los datos (valores) correspondientes sería de la siguiente manera:

## EJERCICIOS

### 1. Autobús – Paradero



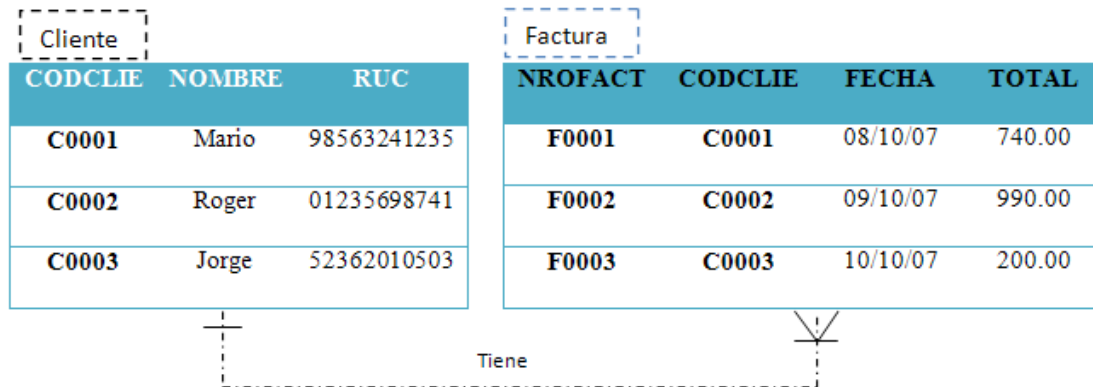
### 2. Alumno - Curso



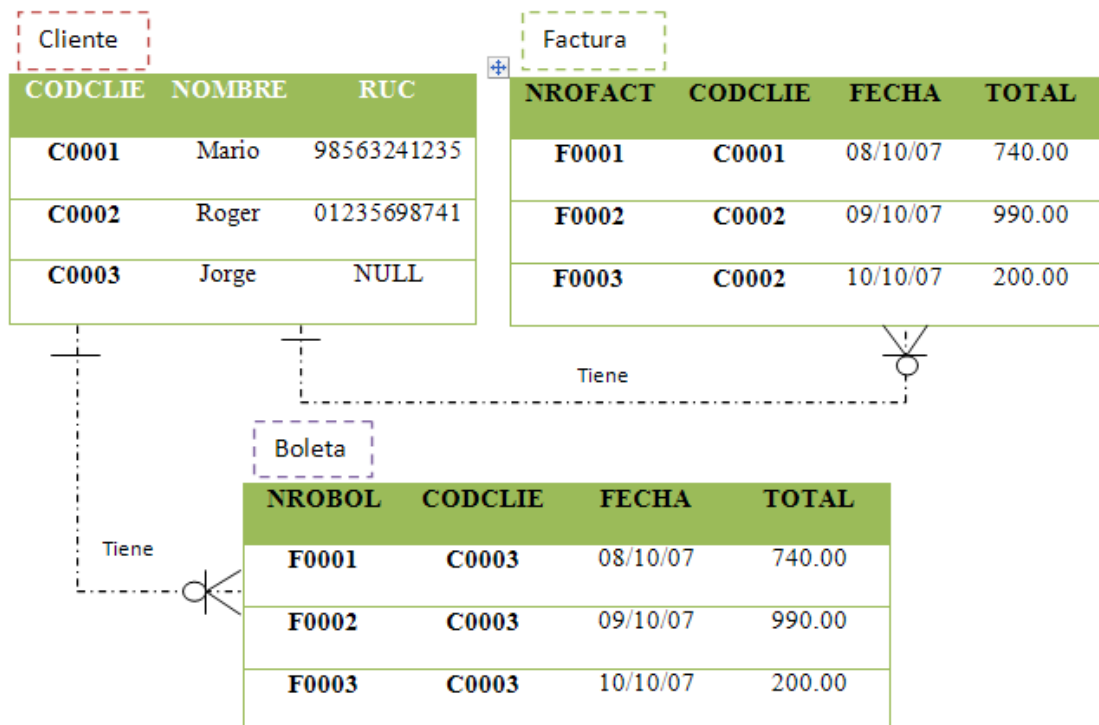
## EJERCICIOS

Interpretar los siguientes casos.

Se tiene como política de ventas de nuestra tienda, vender sólo a clientes jurídicos, es decir a aquellos que cuentan con número de ruc, se les facturará por cada compra que los clientes nos realicen.



El caso anterior muestra una relación Uno a Muchos entre Cliente y Factura ya que la regla del negocio así lo dispone, ahora veamos el caso de una tienda que venda no solo a clientes con ruc sino también a personas naturales (no jurídicos), a estos se les factura con boletas de venta.



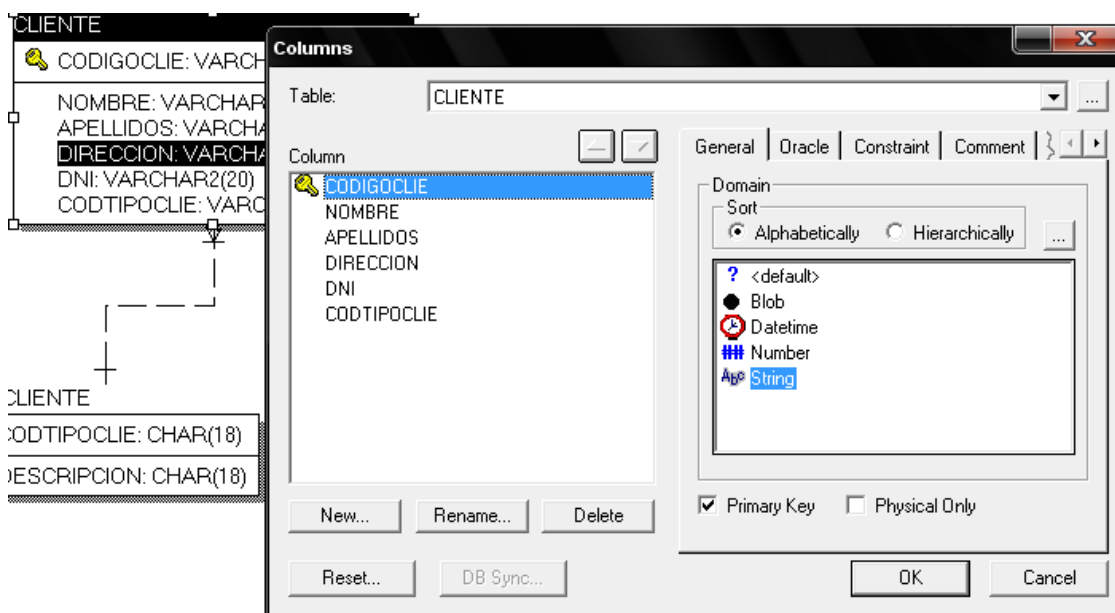
## LABORATORIO # 3

### ✓ CREANDO EL MODELO FÍSICO EN ERWIN

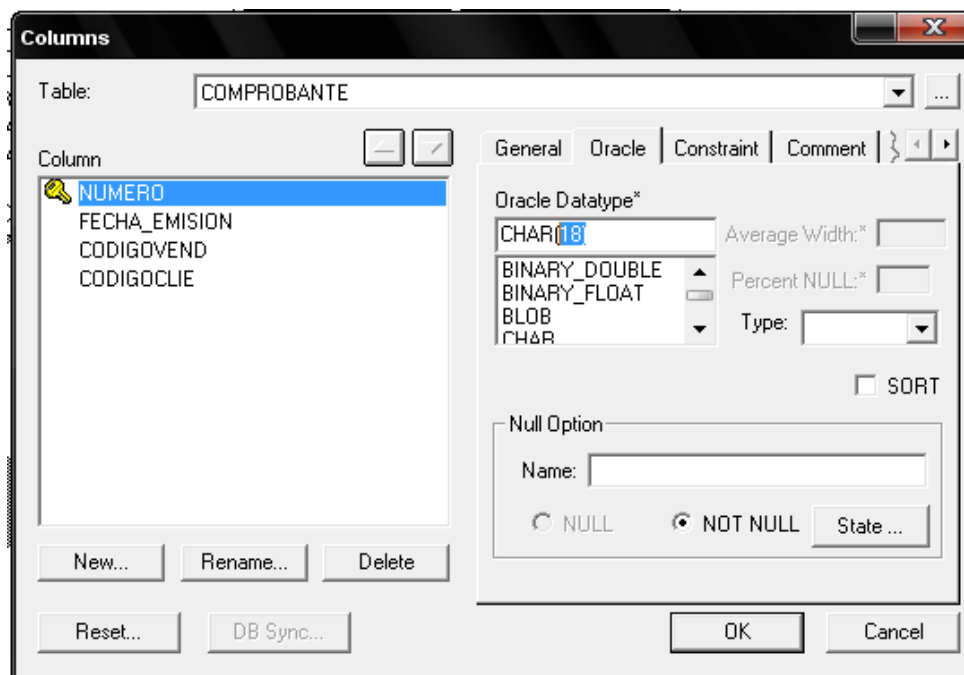
Para ello debemos editar los tipos de datos soportados para cada columna de cada tabla, para ello hacemos clic derecho sobre un espacio vacío y seleccionamos la opción Table Display, ahí elegimos Column Datatype.



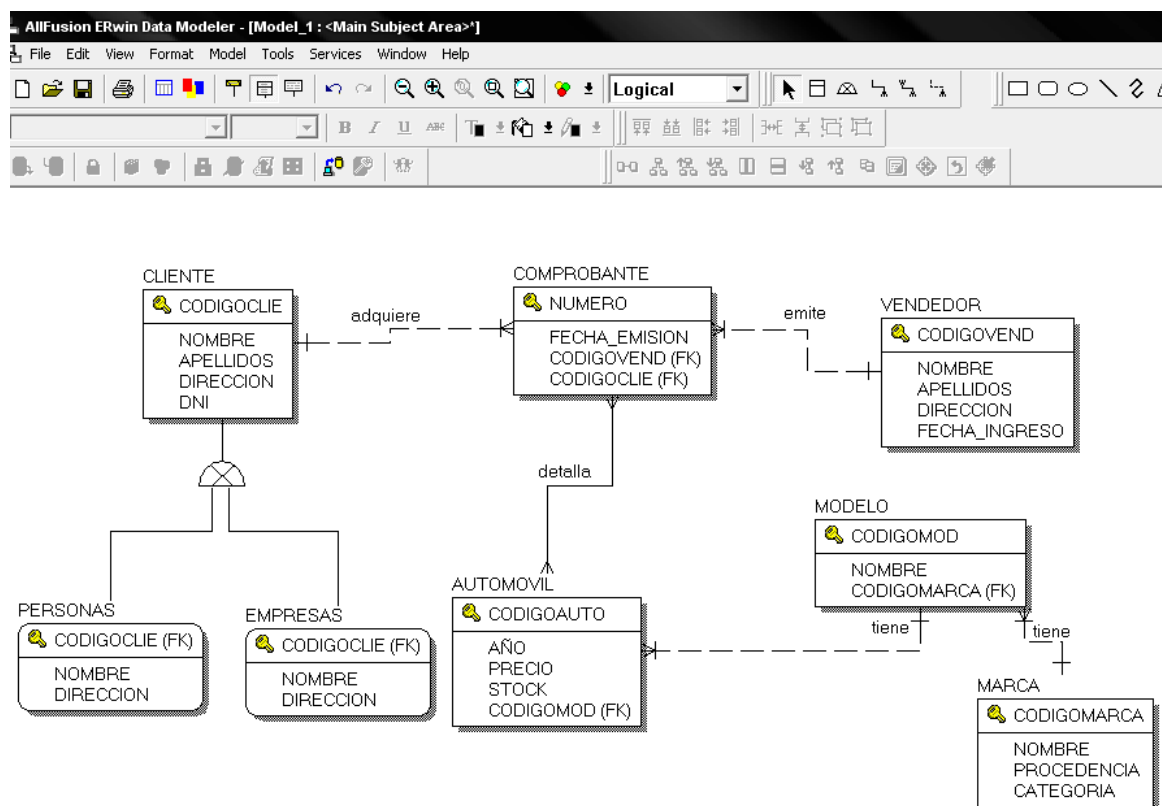
Luego seleccionamos una tabla y hacemos clic derecho, ahí seleccionamos la opción Column con la cual aparecerá la ventana para editar los tipos de datos...



Podemos seleccionar algunos de los tipos de datos estándar (primitivos), o tomar el motor de Base de datos para asignar dominios y extensiones, en la imagen se muestra la pestaña Oracle.

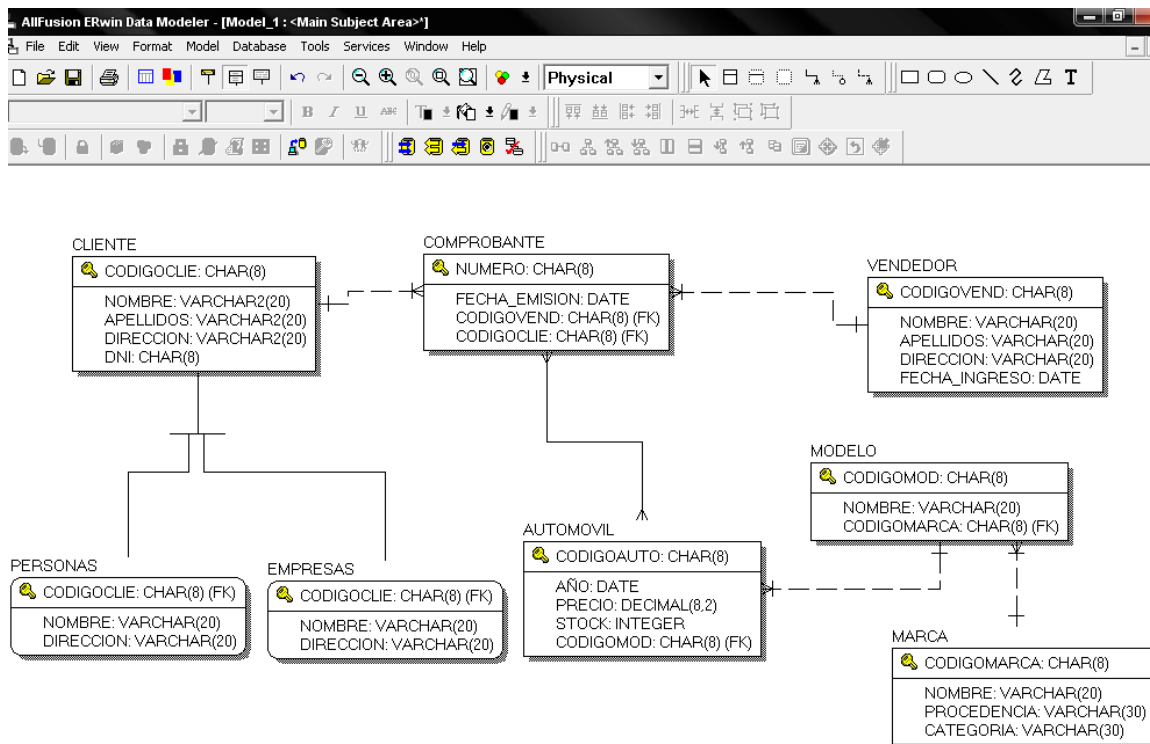


En la siguiente imagen se muestra el modelo Lógico del caso anterior (La concesionaria de automóviles).



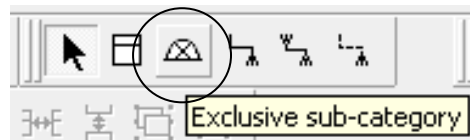


En la siguiente imagen se muestra el modelo Físico del caso anterior (La concesionaria de automóviles), con todos los tipos de datos indicados.

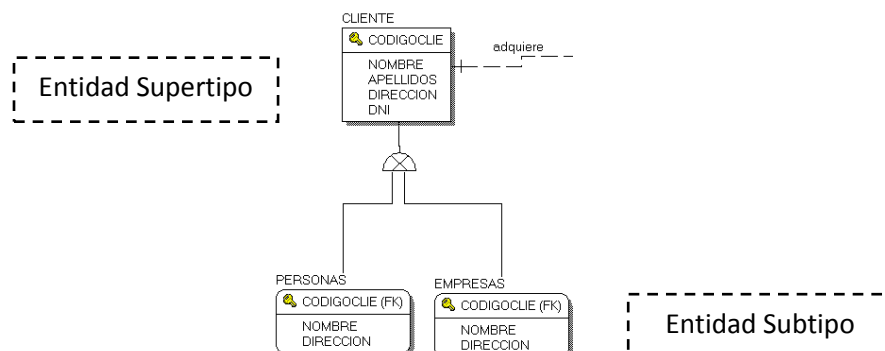


### ✓ CREAR UNA GENERALIZACION JERARQUICA (EN EL MODELO LOGICO)

1. Seleccione el icono de categoría desde el Toolbox.



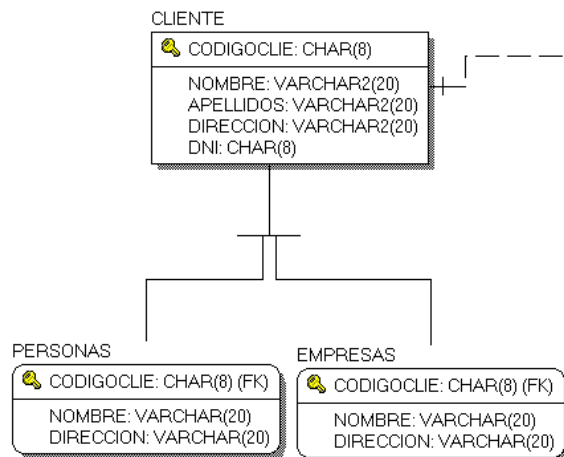
2. Seleccione la entidad Supertipo.
3. Seleccione la entidad Subtipo.
4. Para agregar más entidades hacemos clic en el símbolo de categoría y luego clic en la entidad Subtipo.



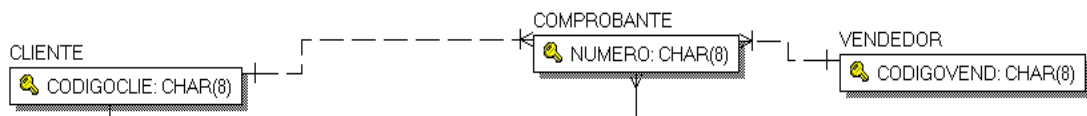
Esto indica que las entidades Subtipos son parte de la entidad Supertipo, en este caso, un cliente puede ser personas naturales o empresas, esto puede indicar distinto trato en las ventas o la aplicación de promociones y/o beneficios dependiendo del tipo de cliente.

## EN EL MODELO FÍSICO

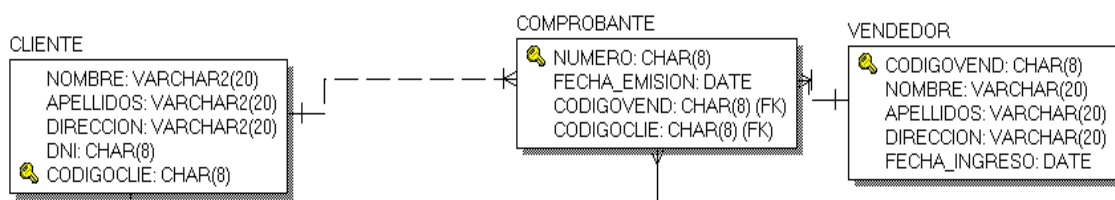
La Generalización se muestra así:



Podemos ver el modelo físico en vista de claves, haciendo clic derecho y seleccionando la opción Table Display, y ahí elegimos Primary key.



Podemos verlo también en vista de columnas, para ello seleccionamos la opción Physical order también desde Table Display.

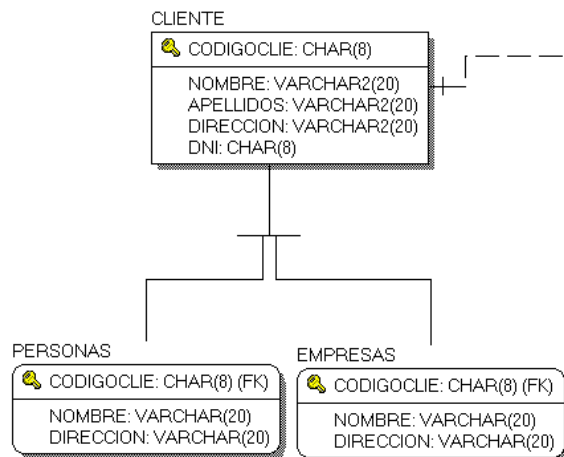


Tanto el Modelo Lógico como el Modelo Físico tienen sus propias opciones para la optimización de los modelos.

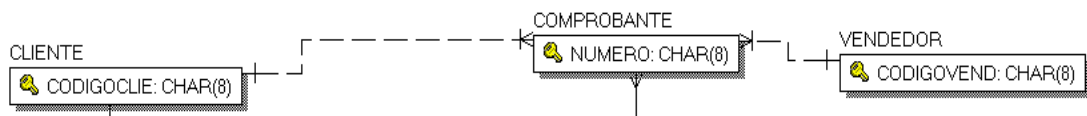
Esto indica que las entidades Subtipos son parte de la entidad Supertipo, en este caso, un cliente puede ser personas naturales o empresas, esto puede indicar distinto trato en las ventas o la aplicación de promociones y/o beneficios dependiendo del tipo de cliente.

## EN EL MODELO FÍSICO

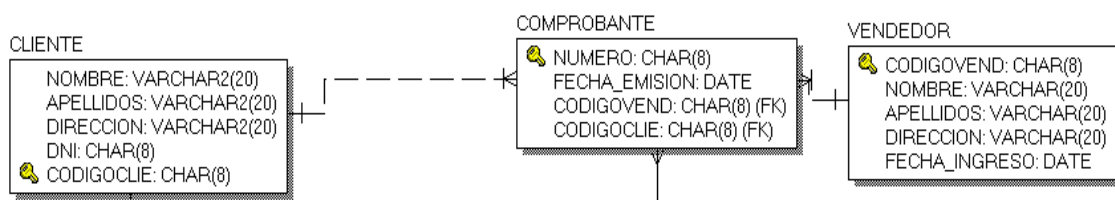
La Generalización se muestra así:



Podemos ver el modelo físico en vista de claves, haciendo clic derecho y seleccionando la opción Table Display, y ahí elegimos Primary key.



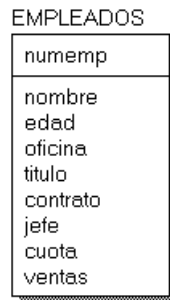
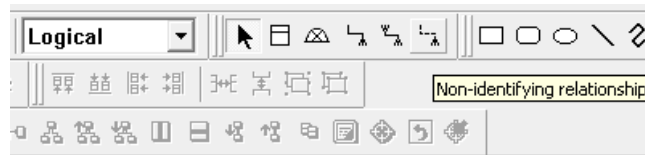
Podemos verlo también en vista de columnas, para ello seleccionamos la opción Physical order también desde Table Display.



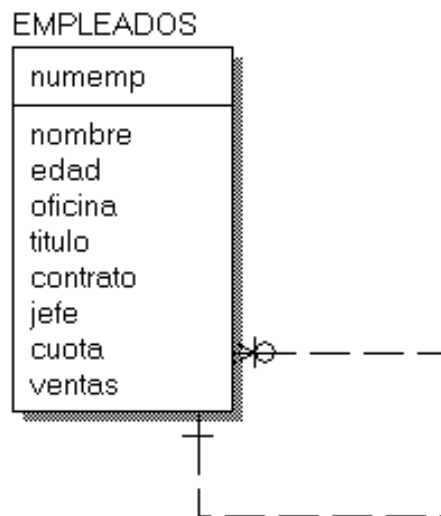
Tanto el Modelo Lógico como el Modelo Físico tienen sus propias opciones para la optimización de los modelos.

### ✓ CREANDO UNA RELACION RECURSIVA

Creamos una entidad en el Modelo Lógico.



Hacemos clic en el icono de la relación Uno a Muchos no identificada, luego hacemos dos veces clic sobre la tabla (no doble clic), con esto tendremos la relación recursiva.



Con la opción Relationship Properties podemos editar la cardinalidad de la relación, así como la frase verbal.

