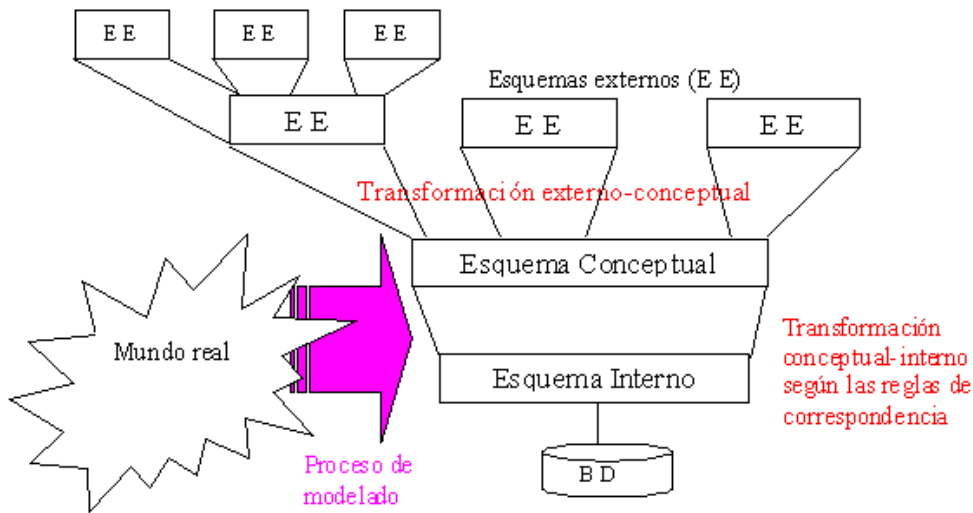




UNIVERSIDAD
DE LOS ANDES



Bases de datos Unidad 3

**Universidad de Los Andes
Escuela de Ingeniería de Sistemas
Departamento de Computación**

Tema 1. Modelo relacional y objeto-relacional

Tema 1. Modelo relacional y objeto-relacional

▶ **Contenido:**

▶ Conceptos básicos

- ▶ Reglas de transformación de ERE o del diagrama de clases UML al objeto-relacional
- ▶ Enfoque por descomposición
- ▶ Normalización
- ▶ Restricciones y reglas de integridad

▶ **Objetivo:**

- ▶ Desarrollar habilidades en el modelado de bases de datos relacional y objeto-relacional

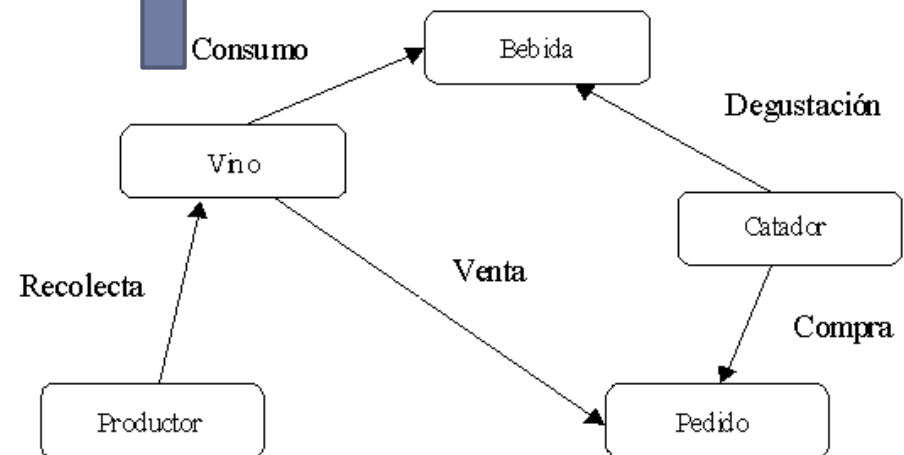
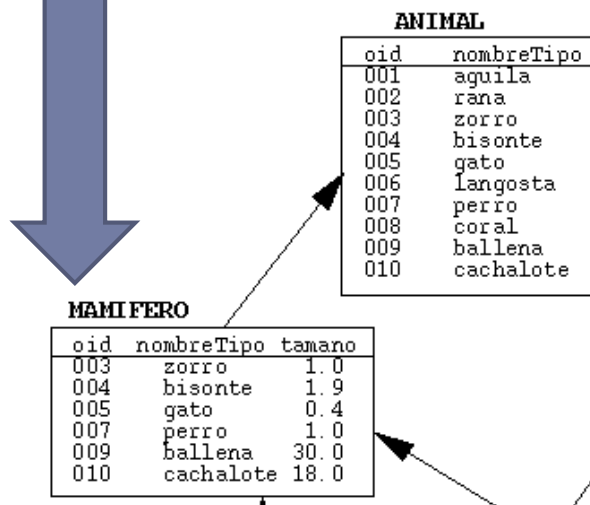
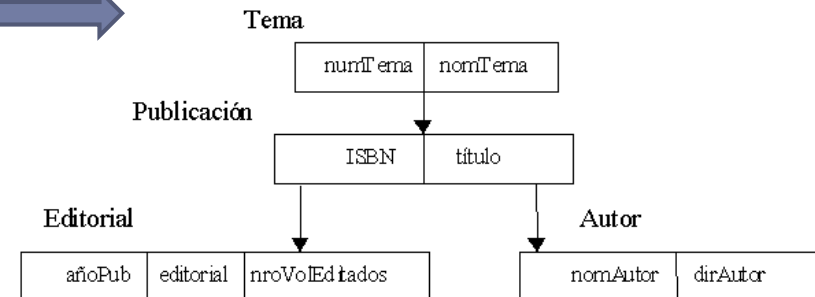
▶ **Actividades:**

- ▶ Leer: Elmasri y Navathe, cap. 5, 7, 10, 11, 16 y 22

Modelos básicos

- ▶ Son los modelos sobre los que se han desarrollado la mayoría de los **SMBD**, estos son:

- ▶ Jerárquico
- ▶ Redes
- ▶ Relacional
- ▶ Orientado por objetos
- ▶ Objeto-Relacional



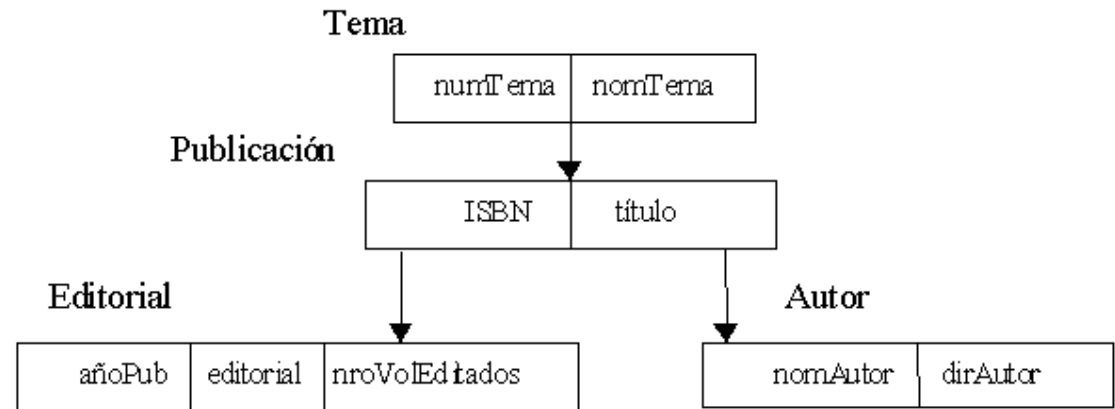
Modelo jerárquico

▶ **Conceptos básicos:**

- ▶ **Campo:** es la unidad de datos que posee un nombre
- ▶ **Segmento:** es una colección de campos consecutivos en la base de datos que posee un nombre y que constituye la unidad de intercambio entre la BD y los PA
 - ▶ Los segmentos están ligados por asociaciones I:N donde un segmento padre tiene N segmentos hijos, bien sea en el ámbito de tipos o de ocurrencias, formando así un árbol de segmentos
- ▶ **Árbol de segmentos:**
 - ▶ colección de segmentos ligados por asociaciones padre - hijos, organizados bajo la forma de una jerarquía
- ▶ Base de datos jerárquica:
 - BD compuesta de un bosque de segmentos
 - se representa con árboles de segmentos cuyos nodos son los segmentos y las aristas indican las asociaciones I:N

Modelo jerárquico

- ▶ **IMS/VS (Information Management System/Virtual Storage) -IBM, primera versión aparece en 1968**



DBD NAME = Publica

SEGM NAME = Tema, BYTES = 44

FIELD NAME = (NumTema, SEQ), BYTES = 4, START = 1

FIELD NAME = NomTema, BYTES = 40, START = 5

SEGM NAME = Publicacion, PARENT = Tema, BYTES = 96

FIELD NAME = (ISBN, SEQ), BYTES = 16, START = 1

FIELD NAME = Titulo, BYTES = 80, START = 17

SEGM NAME = Editorial, PARENT = Publicacion, BYTES = 40

FIELD NAME = (AnioPub, SEQ), BYTES = 4, START = 1

FIELD NAME = Editorial, BYTES = 34, START = 5

FIELD NAME = NroVolEditados, BYTES = 2, START = 39

SEGM NAME = Autor, PARENT = Publicacion, BYTES = 256

FIELD NAME = (NomAut, SEQ), BYTES = 20, START = 1

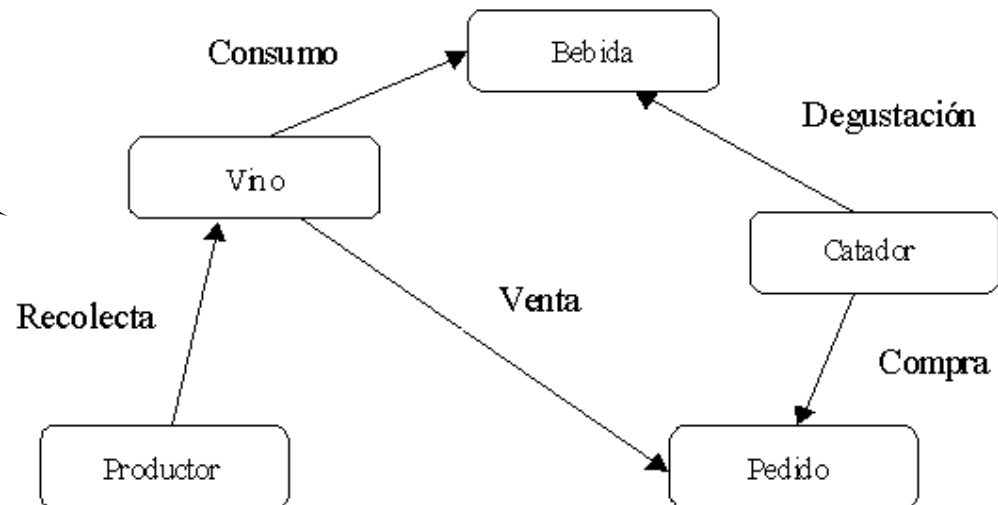
FIELD NAME = Direccion, BYTES = 236, START = 21

- ▶ **Propuesto por el grupo DBTG de CODASYL**
- ▶ **Conceptos básicos:**
 - ▶ **Átomo o item de dato:** es la unidad de datos que posee un nombre
 - ▶ **Agregado de datos:** es una colección de átomos arreglados consecutivamente en la base de datos que posee un nombre
 - ▶ Ellos son de dos tipos:
 - vectores o arreglos unidimensionales y
 - grupos repetitivos
 - ▶ **Registro:** es una colección de agregados y de átomos consecutivos en la base de datos y que constituyen la unidad de intercambio entre la BD y los PA
 - ▶ **Conjunto:** es la asociación entre un registro propietario y n registros miembros
 - ▶ Las limitaciones del modelo hacen que un registro o tipo de registro no pueda ser propietario y miembro a la vez en el mismo conjunto y que una ocurrencia de un registro no pueda pertenecer a varias ocurrencias del mismo conjunto

Modelo de redes

- ▶ **Base de datos en redes:** es una BD compuesta de registros ligados o asociados entre ellos por los conjuntos. Ella se representa a nivel de tipo con un grafo de tipos de registros cuyos nodos son los tipos de registros y las aristas son los tipos de conjuntos orientados del propietario hacia los miembros

Base de datos de una productora de vinos



Ejemplo de esquema del modelo de redes

Schema name is Vinos

Area name is areaVinos

record name is Productor

01 codPro picture is "9(4)"
02 nomPro type is character 32

Record name is Vino

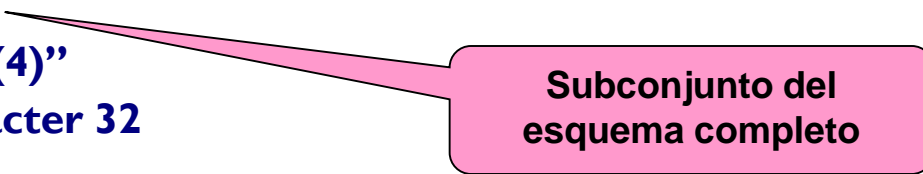
01 codVino picture is "9(6)"
01 nomVino type is character 16
01 Cosecha
 02 dia picture "99"
 02 mes picture "99"
 02 anio picture "99"

Set name is recolecta; order is sorted

Owner is Productor

Member is Vino optional automatic; ascending key is codVino duplicates not allowed

.....



**Subconjunto del
esquema completo**

Modelo relacional (E. Codd en 1970)

- ▶ Está basado en la teoría de normalización de las relaciones, que permite eliminar el comportamiento anormal de las relaciones, luego de actualizaciones, así como el control de la redundancia de datos
- ▶ **Conceptos básicos**

- ▶ **Dominio:** conjunto de valores

$D1 = \{\text{'rojo'}, \text{'verde'}, \text{'azul'}\}$

$D2 = \{\text{'fiat'}, \text{'toyota'}, \text{'ford'}, \text{'honda'}\}$

- ▶ **Relación:** subconjunto del producto cartesiano de una lista de dominios

$R1 = \{(\text{'fiat'}, \text{'verde'}), (\text{'toyota'}, \text{'azul'}), (\text{'ford'}, \text{'rojo'})\}$

$R2 = \{(\text{'rojo'}, \text{'honda'})\}$

$R3 = \{\}$

R1	D2	D1
	fiat	verde
	toyota	azul
	ford	rojo

Modelo relacional

- ▶ **Atributo:** columna de la relación identificada con un nombre
- ▶ **Esquema de una relación o tabla:** nombre de la relación seguido de la lista de sus atributos con sus dominios

Relación
o tabla

R(marca, color)

Esquema por intensión

Tupla o
fila

Esquema por extensión

Atributo o
columna

R	marca	color
	fiat	verde
	toyota	azul
	ford	rojo

Base de datos relacional (BDR)

- ▶ **Base de datos cuyo esquema es un conjunto de esquemas de relación de diferente nombre cada uno, y donde sus ocurrencias son las tuplas de esas relaciones**
- ▶ **Reglas de formación:**
 1. Cada relación o tabla contiene un solo tipo de fila o tupla
 2. Cada tupla tiene un número fijo de atributos o columnas
 3. No se permiten atributos compuestos o grupos repetitivos
 4. Cada tupla es única y se identifica con su clave primaria
 5. Un atributo o grupo de ellos que identifiquen unívoca e inequívocamente cada tupla de la relación es una clave candidata
 6. La clave primaria de una relación se selecciona entre las claves candidatas

► Reglas de formación

7. Si un atributo A pertenece a $R1$ es también la clave primaria de $R2$, entonces A es un atributo foráneo de $R1$
8. El orden de las tuplas en la relación es irrelevante
9. Los valores de los atributos deben pertenecer al dominio de cada atributo definido en ella
10. Un mismo dominio puede ser usado por diferentes atributos
11. A partir de una o más tablas se pueden producir nuevas tablas diferentes mediante el uso de las operaciones del álgebra relacional

► **Reglas de integridad:**

1. **De la relación:** ningún componente de un valor de los atributos que conforman la clave primaria puede ser nulo
2. **De referencia:** sea **A** la clave primaria de **R1** y también un atributo foráneo de **R2**, entonces para toda tupla de **R2** donde **A** \neq nulo debe existir la tupla correspondiente en **R1**
3. **De los valores de un atributo:** son los predicados definidos por el **ABD** sobre los valores de los atributos usando el lenguaje de definición de datos

Ejemplo de restricciones de integridad

Semestre	codSem	fechaInicio	fechaFin	fechaInscripción	fechaRetiro
	'A-04'	21/6/04	10/12/04	14/6/04	10/11/04
	'B-04'	17/1/05	3/6/05	10/1/05	3/5/05
	'U-05'	19/9/05	10/3/06	12/9/05	12/12/05
	'A-06'	20/3/06	14/7/06	18/3/06	10/6/06

fechaInicio < fechaFin

fechaInscripcion < fechaInicio

fechaRetiro < fechaFin

fechaInicio < fechaRetiro

Modelo relacional

▶ Ejemplo de esquema de una BDR

Cliente(codCli, nombre, balance, limiteCredito, descuento)

Envio(direccion, *codCli*)

Pedido(codPed, linea, *dirEnvio*, *codArt*, cantidadPedida, cantidadEnviada)

Articulo(codArt, nomArt, descripcion)

Inventario(codArt, codPlanta, cantidadExistencia, riesgo)

Las claves primarias están subrayadas y las claves foráneas aparecen escritas en itálicas

Restricción de integridad:

cantidadPedida \geq cantidadEnviada

cantidadEnviada \leq cantidadExistencia

Atributo	Descripción	Dominio
codCli	Código del cliente	Cadena(4)
nombre	Nombre del cliente	Cadena(32) solo letras , . " ' - ' "
balance	Balance actual del cliente	Moneda ≥ 0

Tabla de dominios

Atributo	Descripción	Dominio
limiteCredito	Límite de crédito del cliente	Moneda >0
descuento	Descuento aplicado	Moneda ≥ 0
Direccion, dirEnvio	Dirección del cliente y dirección de envío	Cadena(128) letras, digitos, . , # ' / -
codPed	Código del pedido	Cadena(6)
linea	Línea de pedido	Entero > 0
cantidadPedida	Cantidad pedida	Entero > 0
cantidadEnviada	Cantidad enviada del artículo	Entero > 0
codArt	Código del artículo	Cadena(8)
nomArt	Nombre del artículo	Cadena(64) letras digitos , .
descripcion	Descripción del artículo	Cadena(256) letras , . " ' -
codPlanta	Código de la planta	Cadena(2)
cantidadExistencia	Cantidad actual en existencia	Entero > 0
riesgo	Riesgo de inventario	Entero > 0

Modelo objeto-relacional

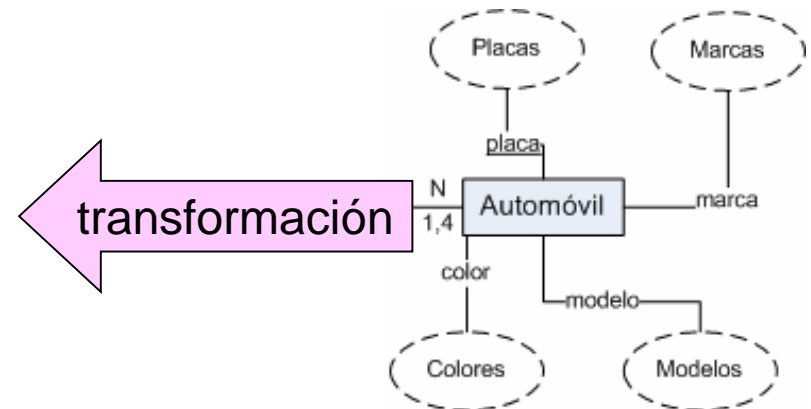
- ▶ **Se pueden crear nuevos**
 - ▶ **Tipos de datos:** pueden ser tipos compuestos, pero que deben ser soportados por el propietario del tipo, esto es debe definir al menos dos métodos transformadores, uno para convertir el tipo nuevo a ASCII y el otro que convierte de ASCII al nuevo tipo. Se soportan tipos complejos como: registros, conjuntos, referencias, listas, pilas, colas y arreglos
 - ▶ **Funciones:** con código en algún lenguaje de programación, por ejemplo: SQL, Java, C, etc.
 - ▶ **Operadores:** asignándole un nombre y asociándolo a una función ya definida o creada con anterioridad
- ▶ **Se soporta el encadenamiento dinámico y herencia en los tipos tupla o registro**
- ▶ **Soporte de reglas a través de los *triggers* o gatillos**

Transformación de modelos de alto a bajo nivel

► Del modelo ERE al relacional

1. Cada conjunto entidad se convierte en un esquema de relación constituido por todos los atributos del conjunto entidad, donde cada tupla en la relación es una entidad en el conjunto entidad con la clave primaria igual a la del conjunto entidad

Automovil(placa, marca, modelo, color)



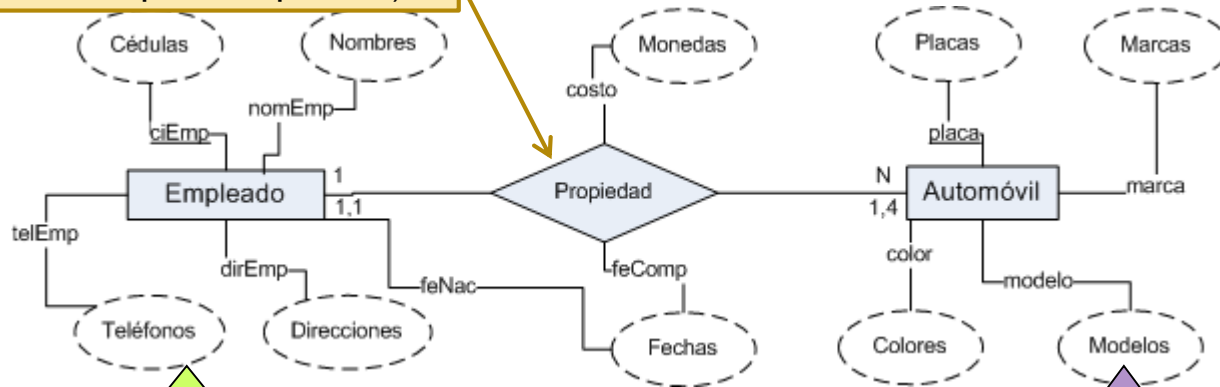
Transformación de ERE al relacional

2. Cada conjunto relación entre los conjuntos entidades que asocia se convierte en un esquema de relación si:

1. El conjunto relación tiene atributos
2. El tipo de correspondencia del mismo es N:M

La clave primaria es la concatenación de las claves primarias de los conjunto entidad que ella asocia y sus atributos son los mismos del conjunto relación tratado

Propiedad(placa, ciEmp, feComp, costo)



Empleado(ciEmp, nomEmp, dirEmp, telEmp)

Automovil(placa, marca, modelo, color)

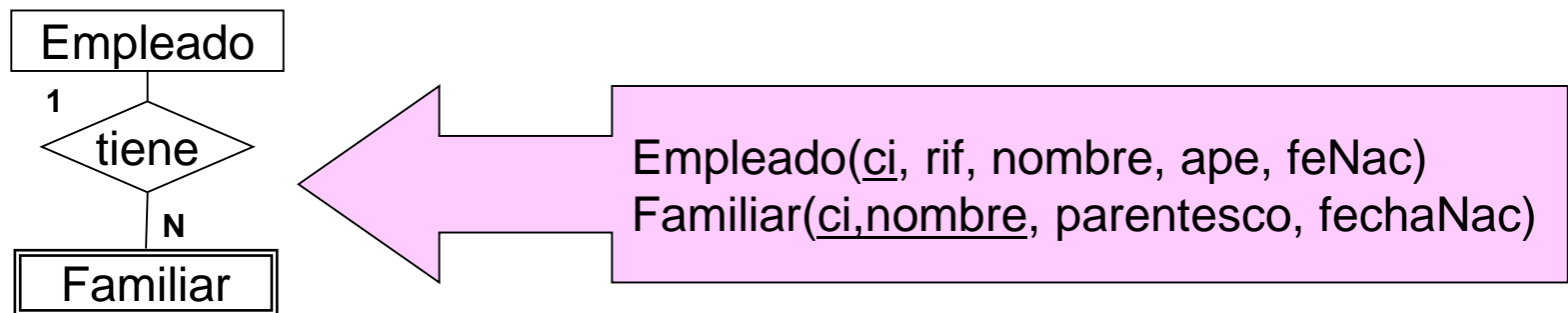
Transformación ERE al relacional

3. Los conjuntos de valores del diagrama ERE se convierten en los dominios del modelo relacional

Atributo	Descripción	Dominio
placa	Placa del vehículo	D1: Placas: Cadena(6) sub(placa, i, 1) ∈ {letras} i=1,3 y sub(placa, j, 1) ∈ {digitos} j=4,5 y sub(placa, 6, 1) ∈ {letras}
marca	Marca del vehículo	D2: Marcas: Cadena(16) sólo letras
modelo	Modelo del vehículo	D3: Modelos: Cadena(16) sub(modelo, i, 1) ∈ {0..9, a..z, A..Z, á,é,í,ó,ú,Á,É,Í,Ó,Ú,ñ,Ñ} i=1,16
color	Color del vehículo	D4: Colores={‘verde’, ‘azul’, ‘rojo’, ‘blanco’, ‘negro’, ‘gris’, ‘naranja’, ‘amarillo’, ‘violeta’, ‘plata’, ‘dorado’, ‘beige’}
ciEmp	Cédulas de identidad	D5: Cedula: Entero > 0
...

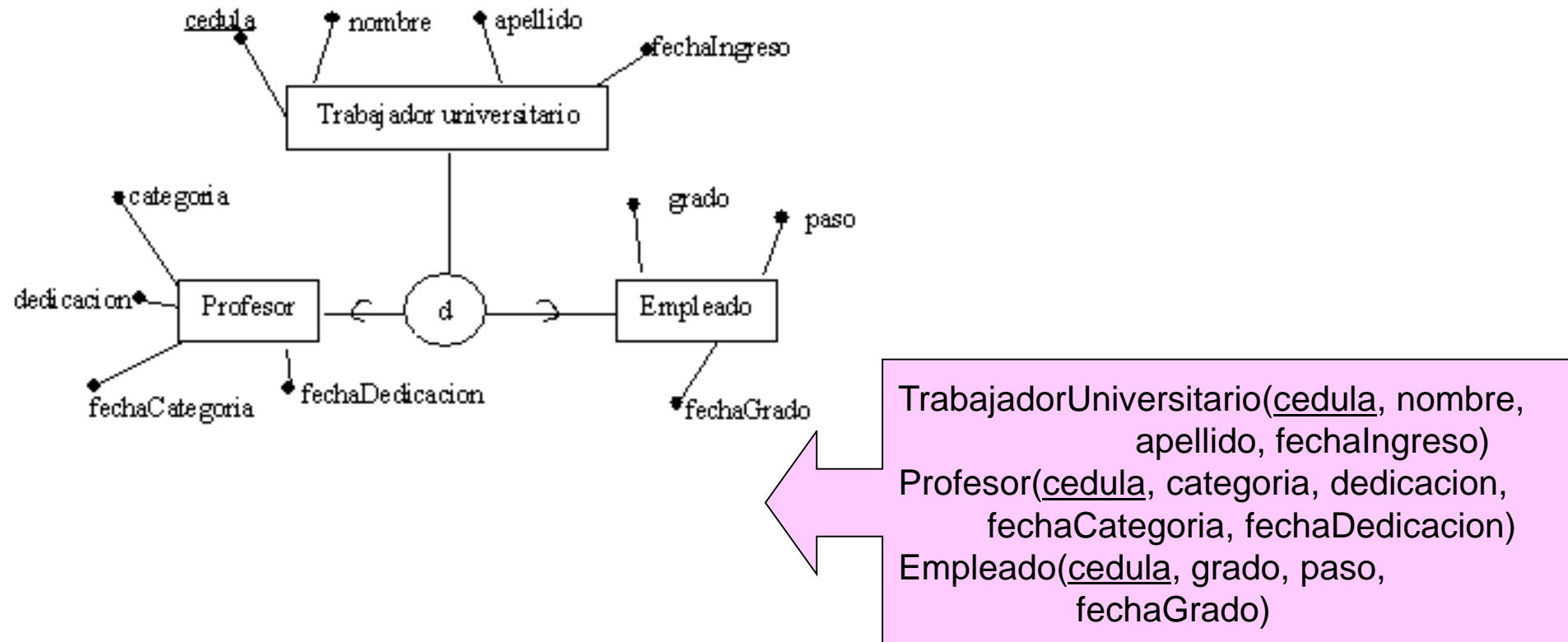
Transformación ERE al relacional

4. Los conjuntos entidad débil se convierten en esquemas de relación con clave primaria igual a la concatenación de la clave primaria del conjunto entidad del cual depende con algún atributo propio del conjunto entidad débil que sirva para identificar cada tupla de la relación



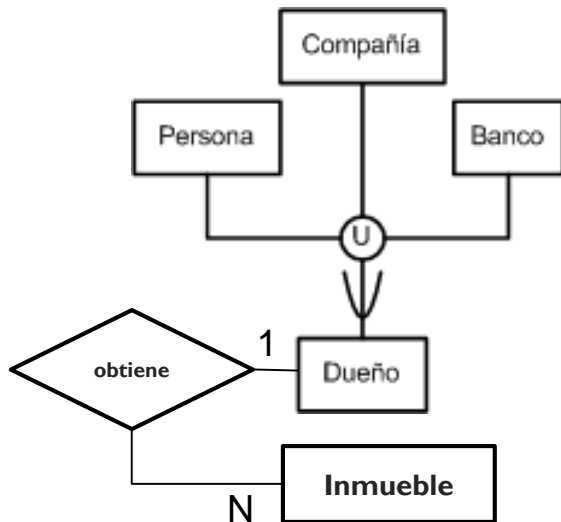
Transformación ERE al relacional

5. Cada especialización es un esquema de relación con los atributos de la especialización y con clave la del conjunto entidad general



Transformación ERE al relacional

6. Una categoría es una subclase de la unión de dos o más superclases, por lo que se crea una clave para la categoría y la misma se coloca en los esquemas de relación de las superclases si ellas tienen diferentes esquemas
7. Los conjuntos relación cuyos tipos de correspondencia sean 1:N o N:1 ó 1:1 indican la toma de la clave primaria del conjunto entidad asociado al 1 para colocarlo como clave foránea en el esquema de relación asociado a N

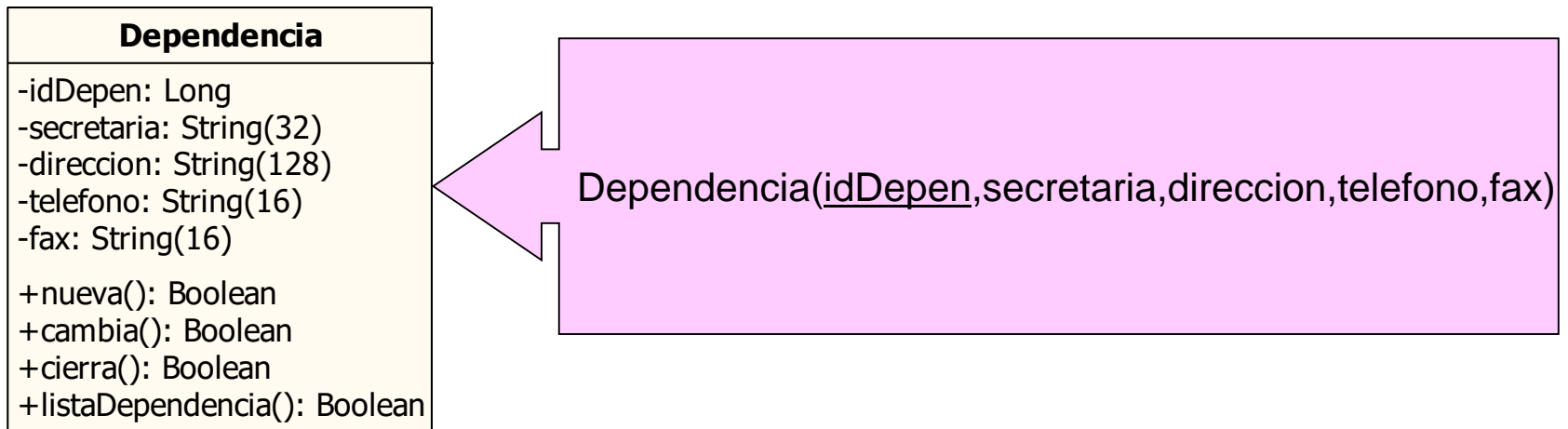


Persona(rif, ci, nombre, ape, feNac)
 Compañía(rif, nombre, fechaFundac, capital)
 Banco(rif, nombre, afiliacion, fechaFundac, capital)
 Dueño(rif, fechaCompra)
 Inmueble(codInm, dir, valorNom, rifDue)

Transformación de modelos de alto a bajo nivel

► Del diagrama de clases UML al Relacional u objeto-relacional

1. Cada clase se convierte en un esquema de relación constituido por todos los atributos de la clase, donde cada tupla en la relación es una instancia de la clase y la clave primaria de la relación se selecciona entre las claves candidatas del conjunto de atributos de la clase

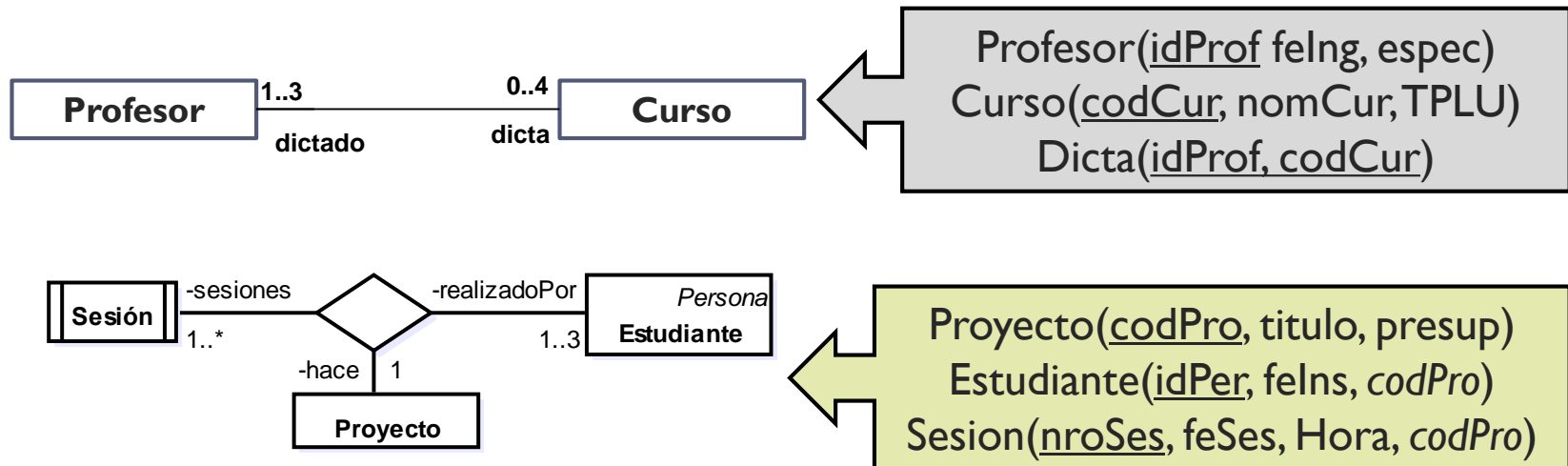


Transformación de modelos de alto a bajo nivel

► Del diagrama de clases UML al Relacional u objeto-relacional

2. Cada relación del tipo asociación, agregación o composición se convierte en un esquema de relación si el tipo de correspondencia del mismo es N:M

La clave primaria es la concatenación de las claves primarias de las clases que ella asocia



Transformación de modelos de alto a bajo nivel

Película(título, año, duracion, tipo)
 Estudio(nombre, fechaFundacion, direccion,
 ciudad, pais, telefonos, dirWeb)
 CompañíaDeProduccion(título, nombre)

Película	
📄 título : Cadena = 'Aún no tiene'	
📄 año : Año = '0000'	
📄 duración : Real = 0.0	
📄 tipo : TiposDePelículas	
📄 nuevaPelícula()	
📄 setTítulo()	
📄 getTítulo()	
📄 setAño()	
📄 getAño()	
📄 setDuración()	
📄 getDuración()	
📄 modificaPelícula()	
📄 despliegue()	

+compañíaProductora **+produce**

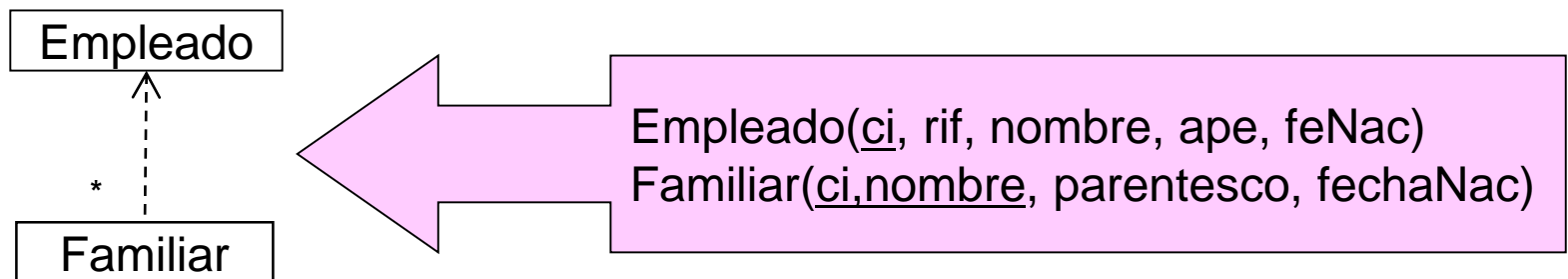
1..* **compañíaDeProducción** **1..***

Estudio	
📄 nombre : Cadena = 'Indefinido'	
📄 fechaFundación : Fecha	
📄 dirección : Cadena = 'Indefinido'	
📄 ciudad : Cadena = 'Indefinido'	
📄 país : Cadena = 'Indefinido'	
📄 teléfonos : List(Cadena)	
📄 dirWeb : Cadena = 'Indefinido'	
📄 nuevoEstudio()	
📄 setNombre()	
📄 getNombre()	
📄 setFechaFundación()	
📄 getFechaFundación()	
📄 setDirección()	
📄 getDirección()	
📄 setCiudad()	
📄 getCiudad()	
📄 setPaís()	
📄 getPaís()	
📄 nuevoTeléfono()	
📄 getTeléfonos()	
📄 setDirWeb()	
📄 getDirWeb()	
📄 modificaEstudio()	
📄 cierreEstudio()	
📄 despliegue()	

Transformación de modelos de alto a bajo nivel

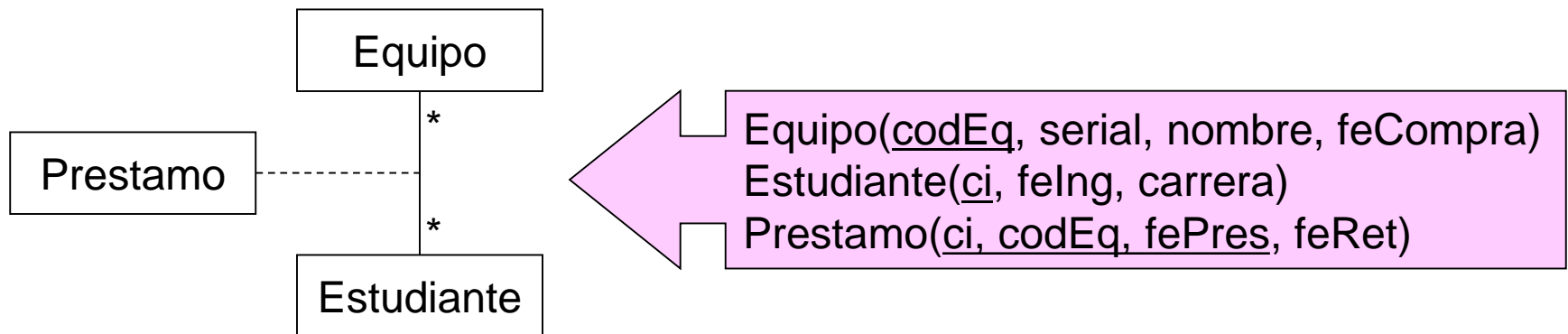
► Del diagrama de clases UML al Relacional u objeto-relacional

3. Las clases dependientes (asociadas con la relación de dependencia) se convierten en esquemas de relación con clave primaria igual a la concatenación de la clave primaria de la clase de la cual depende con algún atributo propio de la clase dependiente que sirva para identificar cada tupla de la relación
4. Las relaciones de agregación y composición solo se convierten en esquemas de relación cuando tienen multiplicidades equivalentes a muchos-a-muchos



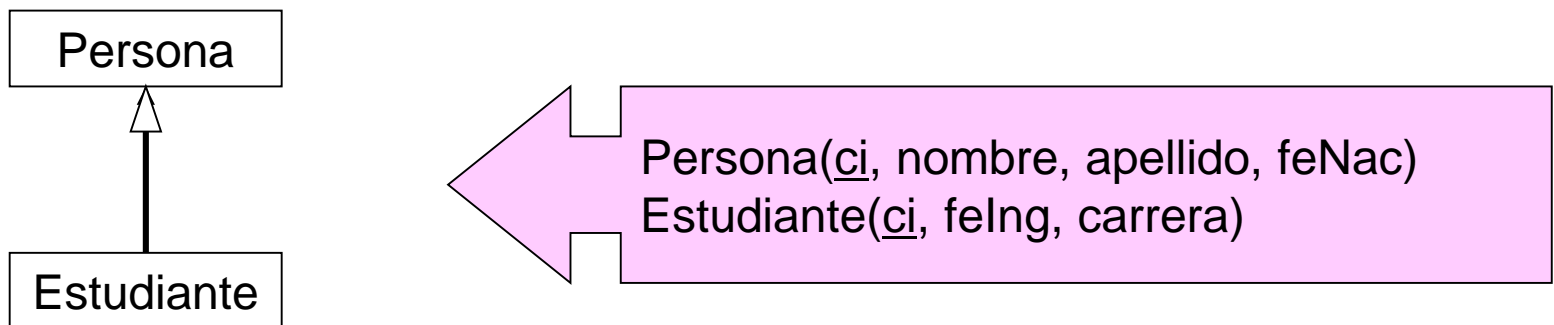
Transformación de modelos de alto a bajo nivel

- ▶ **Del diagrama de clases UML al Relacional u objeto-relacional**
 5. Las clases asociación se convierten en esquemas de relación cuya clave primaria es la concatenación de las claves primarias de las clases asociadas. Si la asociación es M:N se agrega a la clave un atributo que identifique cada tupla de la asociación



Transformación de modelos de alto a bajo nivel

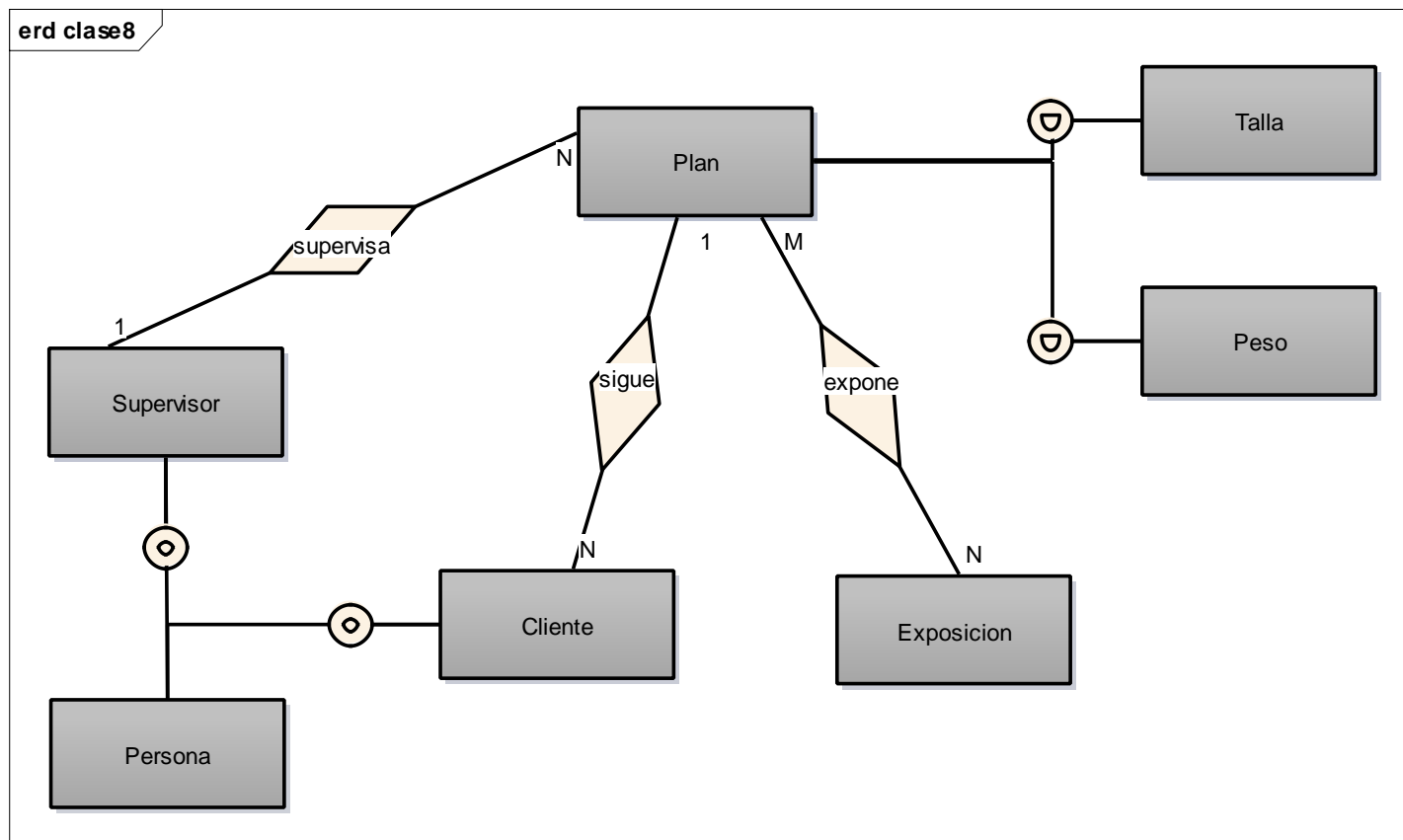
- ▶ **Del diagrama de clases UML al Relacional u objeto-relacional**
- 6. **Las subclases se convierten en esquemas de relación cuya clave primaria es la clave primaria de la superclase**



Autoevaluación

1. **¿Qué se conoce como modelos básicos de BD?**
2. **¿Cuáles son los conceptos básicos del modelo jerárquico?**
3. **¿Qué es una base de datos jerárquica?**
4. **¿Cuáles son los conceptos básicos del modelo de redes y qué es una base de datos en redes?**
5. **¿Cuáles son los conceptos del modelo relacional y qué es una base de datos relacional?**
6. **¿Cuáles son las reglas de formación y las de integridad en BDR?**
7. **¿Qué elementos caracterizan las bases de datos objeto-relacional?**
8. **¿Cuáles son las reglas de transformación del modelo ERE al relacional y de los diagramas de clase UML al modelo relacional?**

- I. Para el diagrama ERE mostrado a continuación obtenga por transformación el modelo relacional correspondiente y agregue la tabla de dominios



Ejercicios

Atributo	Descripción	Dominio
Persona		
identificador	Identificador de la persona. Clave primaria de Persona.	Entero largo = D1
nombre	Nombre de la persona. No puede ser nulo.	Cadena(32) Alf={letras, ', -, /} = D2
apellido	Apellido de la persona. No puede ser nulo.	D2
celular	Número del teléfono celular de la persona. No puede ser nulo.	Cadena(16) Alf={números, -} = D3
Cliente		
direccion	Dirección del propietario	Cadena(128) Alf={letras, números, -, /, ., °, #} = D7
telefono	Teléfono del propietario. No puede ser nulo.	D3
Supervisor		
fechaIniSup	Fecha de inicio de supervisor. No puede ser nulo.	Fecha = D4
comision	Comisión de la supervisión sobre las ventas. No puede ser nulo.	Numérico de 10 dígitos y 2 decimales =D5
especialidad	Especialidad del supervisor. No puede ser nulo.	D2
Plan		
nroPlan	Número del plan. Clave primaria.	Entero+ = D6
nombre	Nombre del plan. No puede ser nulo.	Cadena(64) Alf={letras, números, -, /} = D8
fechaCreacion	Fecha de creación del plan. No puede ser nulo.	D4
pesoInicial	Peso inicial del cliente del plan.	D5
tallaInicial	Talla inicial del cliente del plan.	D5
masaCorporal	Masa corporal del cliente del plan.	D5
Talla		
talla	Talla actual del cliente. No puede ser nulo.	D5
fechaTallaActual	Fecha de la talla actual. No puede ser nulo.	D4
Peso		
peso	Peso actual del cliente. No puede ser nulo.	D5
fechaPesoActual	Fecha del peso actual. No puede ser nulo.	D4
Exposición		
nroExpo	Número de la exposición. Clave primaria de Exposición	D6
fechaInicio	Fecha de inicio de la exposición	D4
fechaFin	Fecha de cierre de la exposición	D4
titulo	Título de la exposición	D2
Restricciones de integridad		
1.	Todas las claves deben ser únicas.	
2.	fechaInicio < fechaFin	
3.	fechaCreacion < fechaInicio	
4.	fechaTallaActual >= fechaCreacion	
5.	fechaPesoActual >= fechaCreacion	

2. Realice la transformación al modelo relacional junto con su tabla de dominios para el diagrama de clases UML de la figura

