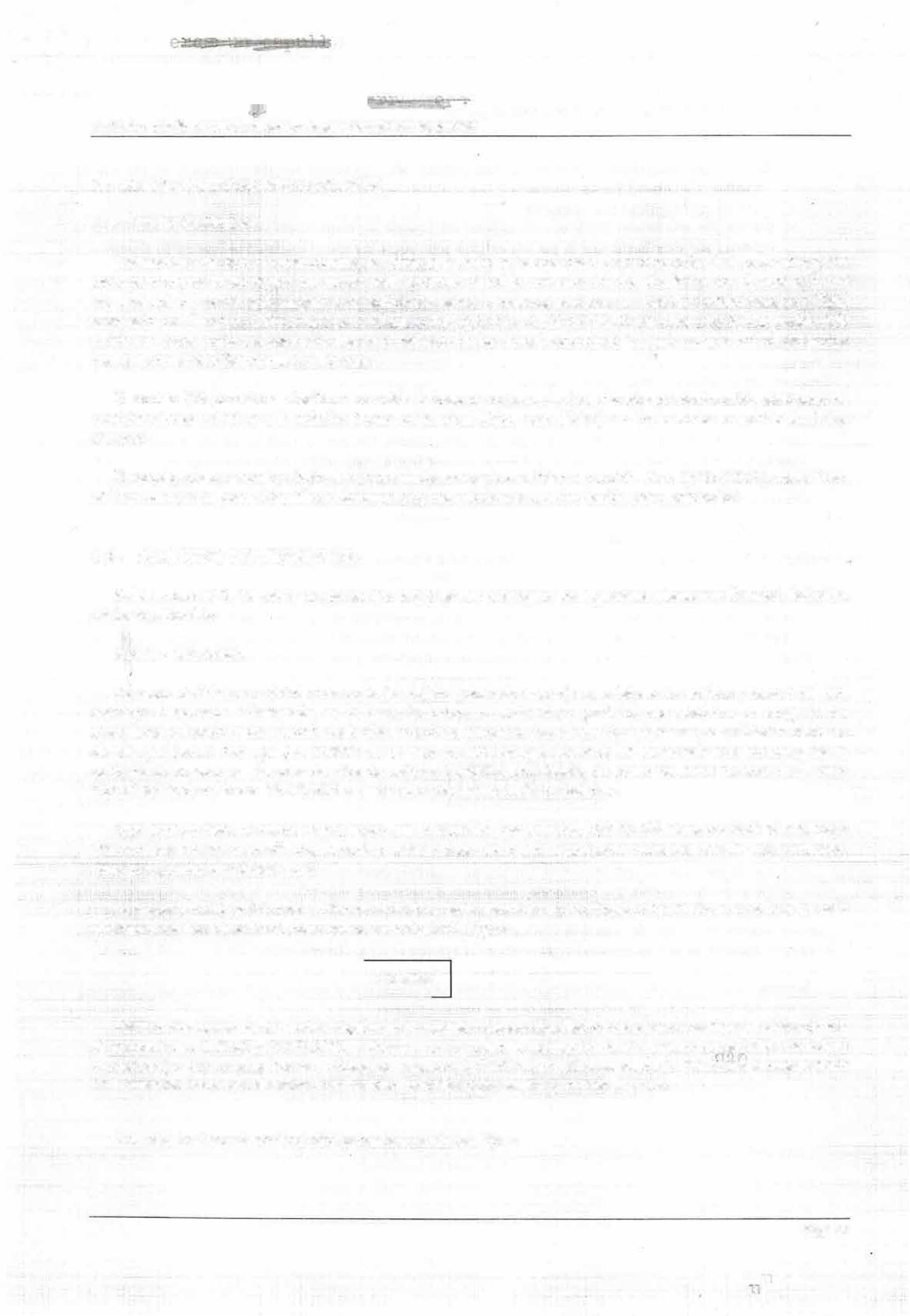
í)



'-

Análisis y diseño detallado de:aplicaciones informáticas de gestión

1. **- MODELO ENTIDAD/RELACION (E/R)**
   1. **- INTRODUCCION.**

Los modelos de datos convencionales no ofrecen la suficiente capacidad de abstracción ni el poder expresivo como para captar la semántica del mundo real, haciendo difícil la comunicación del diseñador con el usuario. Entre los modelos de datos que surgen para solucionar estos problemas destaca el Modelo Entidad/Relación (E,I\_R) propuesto por Peter P. Chen en sus dos artículos CHEN(1976) y CHEN(1977). Según Chen (1976) "el modelo E/R puede ser usado como una base para una vista unificada de los datos", adoptando "el enfoque más natural del mundo real que consiste en entidades e relaciones".

El modelo E/R permite al diseñador concebir la base de datos a un nivel superior de abstracción, aislándolo de consideraciones relativas a la máquina (tanto en su nivel lógico como físico) y a los usuarios en particular (nivel externo)

El modelo, como su nombre indica, se apoya en dos conceptos: entidad e relación. Para CHEN (1976), una entidad es "una cosa que se puede identificar claramente" y una relación es una vinculación entre entidades".

2.2.- ELEMENTOS DEL MODELO E/R.

En el modelo E/R, tal como fue propuesto por Chen, se distinguen los siguientes elementos: Entidad, Relación, Atributo y Dominio.

* + 1. - *ENTIDAD.*

Se puede definir una entidad como cualquier objeto (real o abstracto) que existe en la realidad y acerca del cual queremos almacenar información en la base de datos. La estructura genérica que describe un conjunto de ejemplares aplicando la abstracción de la clasificación se denomina *tipo de entidad* mientras que *entidades* cada uno de los ejemplares. Por ejemplo, CURSO es un tipo de entidad que describe las características comunes de un conjunto de cursos; un ejemplar del tipo de entidad CURSO sería "Diseño de Bases de Datos Relacionales". Otro tipo de entidad podría ser PROFESOR y un ejemplar del mismo sería Sr. Sánchez.

Cada entidad tiene asociado un predicado que la describe. Por ejemplo, PROFESOR tiene asociado el predicado "persona que ejerce o enseña una materia o arte" y cada uno de los ejemplares tiene que cumplir ese predicado, como por ejemplo el profesor Luis.

La representación gráfica de un tipo de entidad en este modelo es un rectángulo etiquetado en cuyo interior está el nombre del tipo de entidad, como vemos en la siguiente figura:

1 Profesor

Existen dos clases de entidades: **regulares o fuertes,** que son aquellas cuyos ejemplares tienen existencia por sí mismos (como CURSO y PROFESOR), y **débiles,** en las cuales la existencia de un ejemplar de¡:,���e de que exista un cierto ejemplar de otro tipo de entidad (por ejemplo, DEPARTAMENTO depende de EMPRESA);'y la desaparición de la empresa hacen que desaparezcan también todas los departamentos de esa empresa.

A su vez, los tipos de entidad débiles pueden ser de dos tipos:

Pág.: 13

Débiles por identificación por lo que una entidad débil no puede ser identificada a no ser que se identifique la entidad fuerte asociada.

- **Débiles por existencia:** por lo que una entidad débil puede ser identificada sin necesidad de identificar la entidad fuerte por la que existe, pero no existirá sino existe la entidad fuerte con la que está asociada.

Una debilidad de identificación implica una debilidad por existencia, pero no a la inversa.

Por ejemplo:

*Empresq,* Identificada por un código de empresa.

*Departamento:* Identificado por el nombre del departamento.

**Identificación:** los departamentos no pueden ser identificados sólo por el nombre, ya que puede haber dos departamentos de distintas empresas con el mismo nombre (por ejemplo departamento de compras).

**Existencia:** Un departamento no existirá sino existe la empresa al cual pertenece. Este tipo de debilidad estará siempre que exista una relación de debilidad por identificación.

*Zona Urbana (barrio)* Identificada por un código de barrio.

*Vivienda* (parcela donde se construye): Identificada por calle y número

Existencia: una vivienda no existe sino existe una zona geográfica en la que se pueda construir. Sin embargo, y si consideramos que no existen calles con el mismo nombre, la debilidad no será por identificación, puesto que es posible identificar una vivienda con independencia de la identificación de la zona urbana en la que se encuentre. Es decir, con el nombre de la calle y el número podremos determinar a que zona urbana pertenece.

Los tipos de entidad débil se representan con dos rectángulos concéntricos con su nombre en el interior, como se puede observar en la figura:

Departamento

#### - RELACIÓN

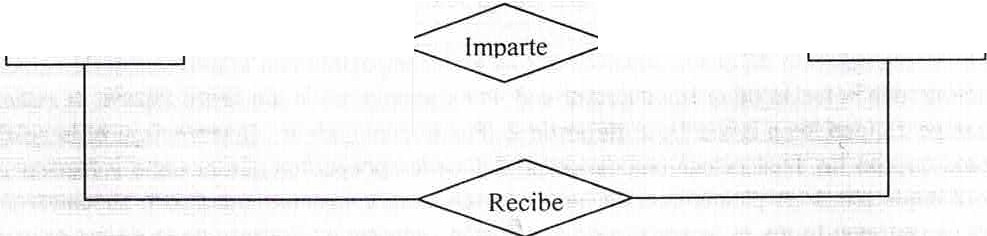
Se entiende por relación una asociación, vinculación o correspondencia entre entidades. Se denomina tipo de relación a la estructura genérica que describe un conjunto de relaciones, mientras que la relación será cada uno de los ejemplares concretos; por tanto el tipo de relación es el resultado de clasificar un conjunto de relaciones. Por ejemplo, **imparte** es un tipo de relación que vincula los dos tipos de entidad **Profesor** y **Curso.** Un ejemplar del tipo de relación **imparte** es la vinculación entre el profesor Sánchez y el curso "Diseño de BD".

Representamos el tipo de relación mediante un rombo etiquetado con el nombre de la relación, unido mediante arcos a los tipos de entidad que asocia, según se muestra en la figura:

. P\_ro\_fi\_es\_o\_r\_

\_;--------<<S;>>-------t. c\_u rs\_o .

Entre dos tipos de entidad puede existir más de un tipo de relación, como se puede observar en la figura:



Profesor

Curso

*2. 2. 2. J.* - *Primera aproximación a la semántica de las relaciones*

El contenido semántico de las relaciones se ha ido completando con conceptos tales como las cardinalidades, lo dependencia en existencia y en identificación, la abstracción de generalización, etc.:

***Elementos de un tipo de relación***

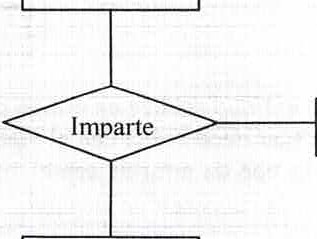
En un tipo de relación se pueden distinguir los siguientes elementos:

* **Nombre:** Al igual que las entidades, los dominios y los atributos, cada tipo relación tiene un nombre que lo distingue unívocamente del resto, mediante el cual ha de ser reverenciado. En la representación gráfica del tipo de relación (un rombo etiquetado) siempre ha de aparecer el nombre, el cual aporta semántica al modelo.
* **Grado:** Es el número de tipos de entidad que participan en un tipo de relación. Así, un tipo de relación es de grado 2 (o binaria) cuando asocia dos tipos de entidad como la relación entre profesor y curso. Un caso particular de relaciones de grado 2 son las reflexivas (también llamadas recursivas), las cuales asocian un tipo de entidad consigo misma: en la figura se muestra el tipo de relación reflexiva **Consta** que asocia TEMA con TEMA, en la que se refleja la posibilidad de que un cierto tema (por ejemplo informática) esté compuesto por subtemas (por ejemplo, ba"ses de datos, sistemas operativos, lenguajes, etc.).

4-------<L- T\_e\_m\_ \_ª ,

Pueden existir también tipos de relación que asocien más de dos tipos de entidad (grado n, n > 2) como en la figura siguiente. En este ejemplo se muestra un profesor con los temas y cursos que imparte.

Profesor

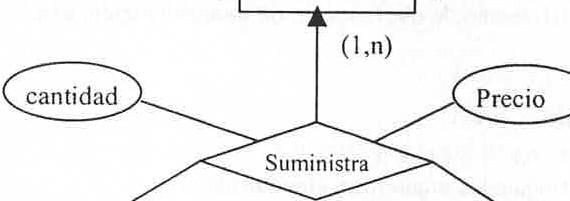


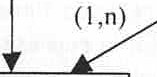
Curso

Tema

Cuando se presenta un tipo de relación de grado n (n >2), es preciso analizar si es propiamente de tal grado, ya que a veces es posible su descomposición en otras de menor grado; mientras que, otras veces, no es posible tal descomposición, ya que la semántica recogida en una y otra solución no es la misma (se pierde semántica).

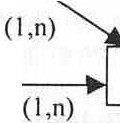
La existencia de una relación de grado superior a 2 no es incompatible con la existencia de relaciones de menor grado\_en las que participen los mismos tipos de entidad. Por ejemplo, en la siguiente figura, la relación de grado 3 **Suministra** coexiste con las tres relaciones de grado 2 (Puede suministrar, Interviene y Necesita), ya que éstas recogen las piezas que puede suministrar un proveedor o para los proyectos que puede suministrar, etc., mientras que la de grado 3 representa las piezas que, de hecho, están siendo suministradas para un cierto proyecto por un determinado proveedor; por tanto, la semántica de la relación ternaria es distinta de la de las relaciones binarias y el usuario podría necesitar que se mantuvieran tres.

Proveedor



(O,n)

(O,n)



fO.n)

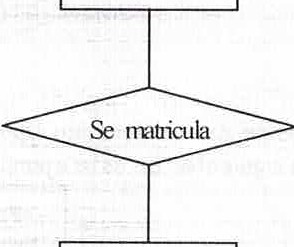
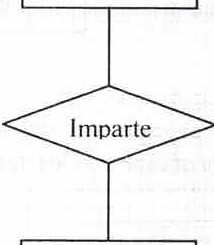
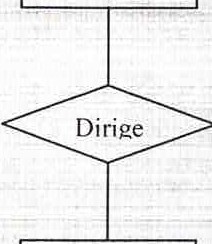
( l,n)

Pie7.a

Proyecto

## (O,n)

- **Tipo de correspondencia:** Es el número máximo de ejemplares de un tipo de entidad que pueden estar asociados, en una determinada relación, con un ejemplar de otro(s) tipo(s); para representarlo gráficamente, bien se pone una etiqueta con 1: 1, l:N o N:M según corresponda al lado de la relación, o bien se orienta el arco de unión en el sentido 1 a N mediante una punta de flecha, tal como aparece en la figura, donde se han incluido.ambos tipos de representación en tres ejemplos de tipos de interpelación.

Profesor Profesor Profesor

1:1 l:N N:M

Curso Curso Curso

- **Papel** ("rol"): Es la función que cada uno de los tipos de entidad realiza en el tipo de relación; se representa poniendo el nombre del papel en el arco que une cada tipo de entidad con el tipo de relación (ver figura). Siempre que no exista ambigüedad se suele prescindir de representar el papel.

-P\_ro\_fi\_e so\_r

\_:------- >-------i. c\_u\_r\_s\_o . es\_impartido imparte

***Cardinalidad de un tipo de entidad.***

Se define como el número máximo y mínimo de ejemplares de un tipo de entidad que pu den estar relacionadas con un ejemplar del otro, u otros tipos de entidad que participan en el tipo de relación. Se representa gráficamente mediante una etiqueta del tipo (O, 1), (1,1), (O,N) ó (l,N), según corresponda, al lado de los tipos de entidades asociados por el tipo de relación, tal como aparece en la figura.

## (1,1) (O,n)

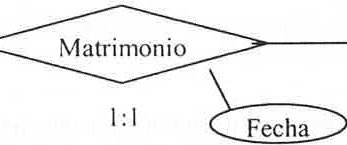
-º-e\_p\_a\_r\_ta\_m\_e\_n\_t\_º .1--------""' ""' 11.-\_P\_ro\_fi\_es\_o\_r\_ \_,

l:N

***Atributos de las relaciones.***

Una relación también puede llevar atributos asociados como muestra la **figura.**

## (O,1)



(O,1)

Hombre

Mujer

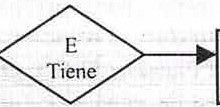
***Dependencia de existencia y de identificación.***

Al igual que existen tipo de entidad débiles y fuertes, también existen tipos de intercalación débiles y fuertes y

dentro de las débiles, por identificación y por existencia según los tipos de entidad que se relacionen.

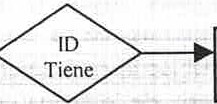
Como vimos, una dependencia por identificación implica siempre una dependencia por existencia, pero no a la inversa.

Si existe dependencia por identificación, el rombo que representa la relación va etiquetado con ID, y con una E en caso de que la dependencia sea en existencia, según se muestra en las figuras:



Empresa

Departam 1



Empresa

Departam 1

*2. 2. 3.* - ***DOMINIO Y VALOR***

Las distintas propiedades o características de un tipo de entidad o de relación toman valores para cada ejemplar de éstas. El conjunto de posibles valores que puede tomar una cierta característica se denomina dominio. Se define dominio como un conjunto de valores homogéneos con un nombre. Para saber si un valor pertenece a un dominio determina o, compro aremos que cumple el predicado que el dominio lleva siempre asociado

Por ejemplo, el valor "inglés" se toma del dominio Idiomas, y cumple el predicado de ser uno de los idiomas posibles del conjunto "español". "inglés". "francés"

El dominio es un elemento del modelo que tiene existencia propia independiente de cualquier otro elemento.

***2. 2. 4.*** - ***ATIBUTO***

Cada una de las propiedades o características que tiene un tipo de entidad o un tipo de relación se denomina atributo; los atributos toman valores de uno o varios dominios.

La representación gráfica de un atributo consiste en cualificar con su nombre el arco que une el dominio con el tipo de entidad o de relación

Sin embargo para simplificar la representación gráfica y siempre que coincida el nombre del dominio con el atributo, será suficiente el círculo u óvalo con el nombre del atributo.

Curso Idiomas O también Curso 1------Q Idiomas

Tipos de atributos. En el modelo E/R se maneJan varios tipos de atributos: *simples, compuestos, monova/uados, mu/t,va/uados, derivados.*

* **At\_cib.utOS-COmpuestos** son los que se pueden d1v1d1r en componentes más pequeños. que representan atributos más básicos con su propio significado independiente. Por ejemplo Dirección se puede dividir en calle. número. piso, código\_postal, y localidad.

Los atributos compuestos pueden formar jerarquías. El valor de un atributo compuesto es la concatenación de los valores de los atributos simples que lo constituyen.

* Los atributos no divisibles se denominan atributos **simples** o **atómicos.**
* Atributos **monoval dos** son a uellos ue tienen un solo valor ara una entidad en particular. Por ejemplo el atributo Edad de la entidad Persona.
* Atributos **multivaluados** son aquellos que pueden tener un con junto de valores para la misma entida,d. Por ejemplo el atributo TTtulos\_universitarios de la entidad Persona.
* En algunos casos se relacionan dos o más valores de atributos; por ejemplo, los atributos Edad y FechaNacimiento de una Persona. Para una entidad Persona en particular, el valor de Edad se puede determinar a partir de la fecha actual y el valor de FechaNacimiento de esa Persona. Por tanto, se dice que el atributo Edad es un atributo **derivado.** Algunos valores de atributos se pueden derivar de entidades relacionadas; por ejemplo, es posible derivar un atributo NúmeroDeEmpleados de una entidad Departamento si se cuenta el número de empleados relacionados con ese departamento a través de la relación trabaja\_en.

1. **3. - RESTRICCIONES**

El modelo E/R tiene como restricción inherente que sólo permite establecer relaciones entre entidades, no estando admitidas entre entidades y relaciones ni entre relaciones. También obliga el modelo a que todas las entidades tengan un identificador, lo que asimismo podría considerarse una restricción inherente. El no tener apenas restricciones inherentes dota al modelo de una gran flexibilidad para la representación del mundo real.

En cuanto a restricciones de integridad, únicamente consideramos las restricciones específicas, distinguiendo entre las restricciones sobre valores y las estructurales.

Las restricciones sobre valores se establecen mediante la definición de dominio, la cual permite limitar los valores del dominio y, por tanto, los de los atributos sobre él definidos, a los de un determinado tipo de datos, o restringirlos a los comprendidos en un rango, o bien declarar los valores posibles.

Las restricciones estructurales se refieren tanto a atributos como a relaciones; estas últimas las analizaremos más adelante cuando tratemos la semántica de las relaciones.

Entre todos los atributos de un tipo de entidad han de existir uno o varios (simples y/o compuestos) que identifiquen unívocamente cada uno de los ejemplares de ese tipo de entidad. Cada uno de estos conjuntos de atributos se denomina *Identificador Candidato* (IC) o Clave Candidata. Cuando un IC es compuesto, el número de los atributos que lo componen debe ser mínimo, en el sentido de que la eliminación de cualquiera de ellos le haría perder su carácter identificador. Luego todo IC debe cumplir la condición de ser unívoco y mínimo. Entre los IC se elige uno como *Identificador Principal* (EP) o Clave Principal y el resto serán Identificadores Alternativos (IA) o Claves Alternativas.

Ejemplo: Un piso será identificado por su **calle, número y su piso** => **IP**

Piso C/Del Hórreo N°13 3º.

Si eliminásemos alguno de los atributos, el piso no podría ser identificado. Para una entidad podríamos tener mas de un identificador:

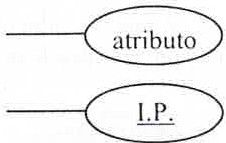
Ejemplo: Socio de un equipo de baloncesto.

**NIF=>IC**

**NUM-SOCIO=>IC**

Podríamos tomar cualquiera de los dos como **IP**

La representación gráfico de estos atributos queda reflejada en lo figuro:

--Q atributo

O también

----•• 1.P.

**2.4.- CONTROL DE REDUNDANCIA**

Es preciso, en los esquemas E/R, analizar lo existencia de redundancias (información repetida), por lo problemas de inconsistencias o los que pueden dar lugar.

Decimos que un elemento de un esquema es redundante cuando puede ser eliminado sin pérdida de semántica (lo que nos dice o representa el modelo no varía).

Existen dos formas principales de redundancia, según el elemento del modelo E/R al que está asociada: redundancia en los atributos (atributos derivados) y redundancia en las relaciones (denominadas también por algunos autores relaciones derivadas).

*2.4.1. Atributos derivados*

Entendemos por atributos derivados (o calculados) aquellos que se obtienen a partir de otros ya existentes, por lo que, aunque son redundantes, no dan lugar a inconsistencias, siempre que en el esquema se indique su condición de derivados y la fórmula mediante la que han de ser calculados.

En lo figura tenemos el atributo número de ejemplares, que puede ser calculado a partir de los ejemplares del libro mediante la relación **tiene.** Para indicarlo gráficamente lo introduciremos en un óvalo discontinuo, almacenando la regla de derivación en el diccionario de datos.

Aunque hay autores que sostienen que dichos atributos no deben ser indicados en este modelo, sino en el lógico (relacionar) o físico, nosotros sí lo vamos a indicar. La implantación o no de dicho atributo en el esquema físico. va a depender de cuestiones de eficiencia, y dependerá del número de actualizaciones frente al de recuperaciones. Así, si es un campo que se consulta mucho, y no está guardado físicamente, sino que es necesario calcularlo, se perdería mucho tiempo. mientras que si el campo está almacenado físicamente y se actualiza es necesario modificarlo constantemente.

N\_ejem . L\_ib\_,\_·o----', --<cG-> ,L!:::::=EJ=e·=m=p=l=a=r==J

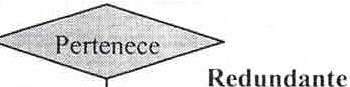
*2.4.2-. Relaciones redundantes.*

Se dice que una relación es redundante cuando su eliminación no implica pérdida de semántica (significado) porque existe la posibilidad de realizar la misma asociación de ejemplares por medio de otras relaciones.

Es condición necesaria, aunque no suficiente, para que una relación sea redundante que forme parte de un ciclo, por lo que hay que estudiar detenidamente los ciclos en el diagrama E/R.

En el ejemplo de la figura se da un ciclo entre PROFESOR, CURSO y DEPARTAMENTO por lo que en principio es posible que aparezca alguna relación redundante. **Supongamos que un profesor sólo puede impartir cursos de doctorado que están adscritos al departamento** al **que él pertenece;** un profesor siempre va o impartir un curso como mínimo (l,n); en este caso. si se conocen los cursos de doctorado que imparte un profesor y el departamento al que está adscrito cado curso. se deduce inmediatamente a qué departamento pertenece dicho profesor; de formo análoga, dado un departamento, si sabemos qué cursos de doctorado tiene adscritos y los profesores que importen dichos cursos, conoceremos qué profesores pertenecen a dicho departamento. por lo que la relación **pertenece**

entre las entidades PROFESOR y DEPARTAMENTO es redundante, su eliminación no produce pérdida de información.



Profesor ( 1,n) ( l ,n)

( 1,n)

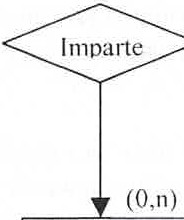
Curso ( 1.11)



( 1, 1)

>---- ('-I:....1'.:\_)---l Dcparta1rn.:11to

En lo siguiente figuro, o pesar de que también existe un ciclo, no hoy ninguno relación redundante En este ejemplo lo semántico es distinto y un departamento puede no tener adscritos cursos de doctorado; además un mismo curso puede estor adscrito o distintos departamentos y puede haber profesores que no importan ningún curso. Lo relación **pertenece** no puede deducirse en este coso de los otros dos, y que aunque sepamos los cursos que ha impartido un profesor y los departamentos a los que están adscritos dichos cursos, no podemos saber a qué, departamento en concreto pertenece dicho profesor; tampoco se tiene esta información poro los profesores que no importen ningún curso. La relación **imparte** tampoco es redundante, yo que un curso de doctorado puede ser impartido por diversos departamentos a codo uno de los cuales pertenecen varios profesores, por lo que no se puede saber qué profesor en concreto importe un determinado curso. Por último, lo relación **adscrito** tampoco es redundante. ya que un curso impartido por un profesor no tiene por qué estar necesariamente adscrito al departamento al que pertenece dicho profesor: hay departamentos que no tienen cursos adscritos y los profesores de estos departamentos pueden colaborar en cursos adscritos a otros departamentos distintos del suyo.

 Profesor



( 1,n) ( 1,n)

( 1.1)



Curso

(O,n)

( 1.n)

Departamento

En resumen, paro que uno relación pueda ser eliminada por redundante se tiene que cumplir:

o) Que exista un ciclo.

b) Que los relaciones que componen el ciclo sean equivalentes semánticamente.

e) Que se puedan asociar los ejemplares de las dos entidades que estaban relacionadas. aún habiéndose eliminado lo relación.

d) Que la relación o bien no tengo atributos o bien éstos puedan ser transferidos o otra a fin de no perder su

semántico.

### - GENERALIZACIÓN/ESPECIALIZACIÓN

En el modelo E/R básico propuesto por CHEN (1976) no se encontraba este tipo de abstracción que fue introducido en posteriores extensiones del modelo (EEÍR: Entidad Relación Extendido). La jerarquía de generalización/especialización, en el modelo E/R, se considero como un caso especial de relación entre varios tipos de entidad (subtipos) y un tipo más general (supertipo) cuyos características son comunes a todos los subtipos. Lo relación que se establece entre los subtipos y el supertipo corresponde o lo noción de "esun" o más precisamente "es\_un\_tipo\_de".

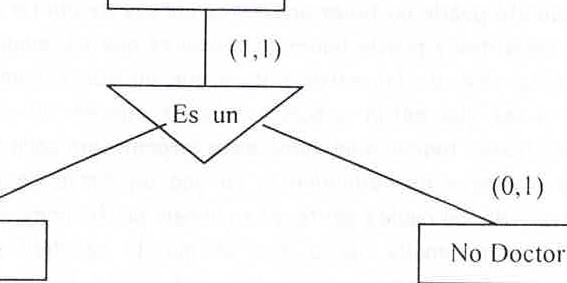
Aunque existen distintas convenciones para representar estas jerarquías de generalización/especialización, se utilizará un triángulo cuyo base es paralela al rectángulo que representa la entidad del supertipo al cuál está conectado; triángulo que también se une o los subtipos, tal como se muestra en lo siguiente figuro.

Esto clase de relación tiene la característico de que todo ejemplar de un subtipo es también un ejemplar del supertipo, aunque no sucede lo contrario, con lo que los cardinalidades serán siempre (l. 1) en el supertipo y (O, **1)** en los subtipos.

Profesor Supertipo

* + - Subtipos

La aparición de estas jerarquías en el modelado de bases de datos puede surgir de dos formas distintas:



(O,1)

Doctor

1. **Generalización.** Se observa que dos o más tipos de entidad comparten varios atributos y/o tipos de relación, de donde se deduce la existencia de un tipo de entidad de nivel superior (supertipo) que contiene los atributos y los tipos de relación comunes a todos los subtipos.
2. **Especialización.** Se observa que un tipo de entidad tiene ciertos atributos y/o tipos de relación que tienen sentido para unos ejemplares pero no para otros, por lo que es conveniente definir uno o varios subtipos que contengan estos atributos y/o tipos de relación específicos, dejando en el supertipo los que son comunes.

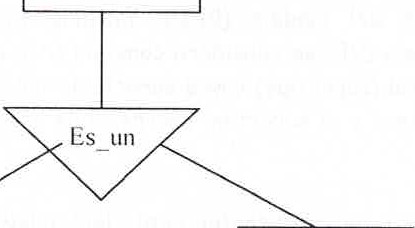
Por tanto, si nos movemos de los subtipos hacia el supertipo, se trata de una generalización; mientras que s1 primero identificamos el supertipo y, a partir deél, llegamos a los subtipos, se trata de una especialización.

Puede ocurrir que se formen, por generalización y/o especial1zac1ón, Jerarquías a más de un nivel donde un subtipo es, a su vez, supertipo de otros.

Persona



Profesor Estudiante



Funcionario Interino

Otra característica muy importante de esta clase de relaciones es la herencia, ya que, en principio todo atributo del supertipo pasa a ser un atributo de los subtipos; por ejemplo, en la jerarquía de la figura anterior tanto los doctores como los no doctores son (o son tipos de) profesores, por lo que heredarán todos los atributos de PROFESOR (Código, Nombre, DNI, Dirección, etc.). Esta característica la diferencia de la clasificación, donde los

subtipos son ejemplares por lo que al heredar los atributos del supertipo lo hacen tomando valores para cada uno de los atributos heredados, mientras que en la generalización propiamente dicha se heredan los atributos, pero sin sus valores.

En este tipo de abstracción los atributos comunes a todos los subtipos (incluidos los identificadores) se asignan al supertipo, mientras que los atributos específicos se asocian al subtipo al cual pertenecen. Del mismo modo, las relaciones que afectan a todos los subtipos se asocian al supertipo, dejándose para los subtipos las relaciones específicas en las que sólo participa el correspondiente subtipo.

La división en subtipos (especialización) puede venir determinada por una condición predefinida (por ejemplo, en función de los valores de un atributo) en cuyo caso se representará la condición (o el atributo discriminante) asociada al triángulo que representa la relación. Si no interesa considerar ninguna condición predefinida, deber ser el usuario, en el momento de insertar un ejemplar en la base de datos, quien especifique a cual de los subtipos pertenece.

La abstracción de generalización/especialización puede ser de cuatro tipos dando lugar a significados semánticos distintos.

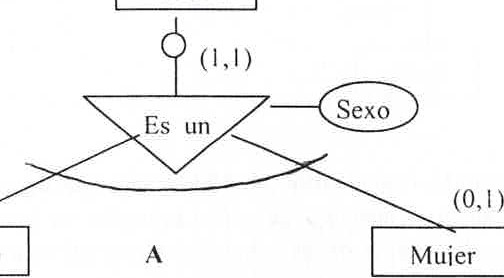
* 1. total exclusiva

c) total no exclusiva

* 1. parcial exclusiva

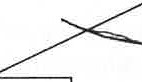
d) parcial no exclusiva

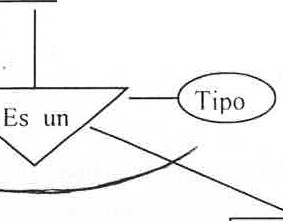
Persona 1 Enfennedad 1



(O,1)

Hombre

(O,1)

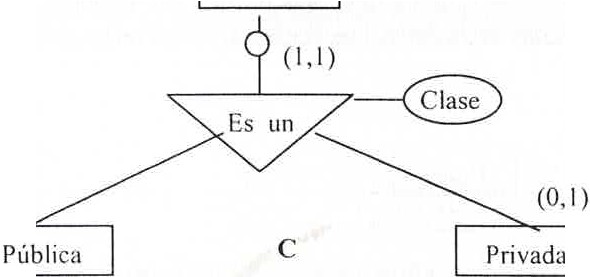


( 1,1)

(O,1)

Vírica B Bacteriana

Emoresa Persona



(O,1)

En el caso A) indicamos que una persona debe de ser Hombre o Mujer, pero no las dos a la vez (exclusividad).



(O,1)

Trabajador

o

En el caso B) es una especialización parcial exclusiva, en la que el tipo de entidad enfermedad (supertipo) se ha especializado en dos subtipos (vírica y bacteriana), de tal forma que un ejemplar de enfermedad, o bien pertenece al

tipo de entidad vírica o bien al tipo de entidad bacteriana o bien al tipo de entidad enfermedad, ya que pueden existir entidades enfermedades las cuales no pueden ser clasificadas en ninguno de los dos subtipos, debido al desconocimiento del valor del atributo tipo utilizado como discrim-inador.

En el caso C) establecemos que toda empresa o bien es pública o privada o de los dos tipos a la vez.

En el caso **D)** establecemos que una persona puede ser un trabajador y/o un estudiante y que existen personas que no pueden clasificarse en ninguno de estos dos subtipos.

* 1. - **AGREGACIÓN.**

La agregación es una abstracción que permite representar tipos de entidad compuestos que se obtienen por unión de otros más simples.

La agregación compuesto/componente, como su propio nombre indica, es una abstracción que permite representar que un todo se obtiene por la unión de diversas partes que pueden ser tipos de objetos distintosy que desempeñan diferentes papeles en la agregación. Por ejemplo, en la siguiente figura, un coche puede verse como la unión del chasis, el motor y las cuatro ruedas.



*(* l. 1)

*f* 1. 1 \

*(-1* 4)

Coche

Chasis Motor Rueda

La agregación miembro/colección es la abstracción que permite representar un todo como una colección de partes, donde todas las partes son de un mismo tipo y desempeñan el mismo papel. Por ejemplo, en la siguiente figura se puede observar cómo un bosque es un todo formado por la agregación de árboles; cada árbol es una parte, pero todos ellos son de un mismo tipo y desempeñan el mismo papel.

1 Bosoue K)------------: A\_rb\_o\_·1

En la agregación miembro/colección a veces se desea establecer un orden entre esas partes. Por ejemplo: una flota se compone de barcos, pero cada uno tiene un orden determinado en la flota. Los barcos se ordenan en la flota según el atributo Num\_de\_barco.

fn l .112\

Flota K) -- \_8\_a\_rc\_o\_

**:orden por**

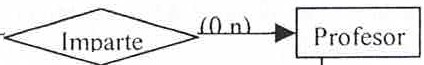
Una lim1tac1ón del modelo E-R es que no es posible expresar relaciones entre relaciones. Se podría solucionar mediante una agregación, creando un tipo de entidad compuesto por un tipo de relación y los tipos de entidad vinculados por la misma, de forma que este nuevo tipo de entidad se pueda relacionar con otros.

### - LA DIMENSIÓN TEMPORAL EN EL MODELO E/R

Es indudable la necesidad de establecer un método semántica y gráfico que recoja de algún modo, en el esquema conceptual, el transcurso del tiempo y su influencia en la forma en que varían los datos. La aproximación más simple la constituyen atributos de tipo fecha que aparecen asociados a algunas entidades como muestra la siguiente figura:

## Proresor

r- nac

l·N

F edic

Por otro lado, podemos analizar si los datos que se pretenden almacenar van a constituir una base de datos histórica o, si por el contrario. sólo nos interesa el estado actual de los mismos. La diferencia entre estos tipos de esquemas se puede apreciar en la siguiente figura donde la parte superior se refiere a los préstamos actuales de libros en una biblioteca. de forma que una vez finalizado el préstamo la correspondiente información desaparece de la base de datos, sin que exista fichero histórico. En la parte inferior se representa el esquema conceptual de todos los préstamos que se han realizado en la biblioteca. recogiendo, además. el periodo de tiempo que duró el préstamo.

En caso de tratarse de datos históricos, los tipos de entidad o de relación correspondientes tendrán asociados siempre atributos de tipo fecha. Para sucesos puntuales, es decir, sin duración, bastará con un solo atributo de este tipo, mientras que para poder almacenar hechos que transcurren en un periodo de tiempo determinado necesitaremos una fecha-1n1c10 *y* una fecha-f,n.

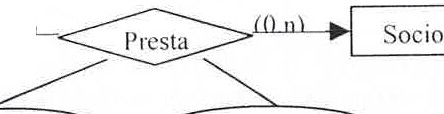
En las bases de datos históricas en las que una relación entre dos ejemplares concretos se pueda repetir en el tiempo, el atributo fecha será multivaluado (se repite), como ocurre en la parte inferior de la figura. donde el mismo ejemplar se puede prestar al mismo socio en repetidas ocasiones.

1:N

1 Eiemolar l ◄ll 5) (Q 1) Socio

r- préstamo

·M



## Eiemolar I◄<1 n)

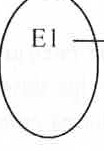
F préstamo F devolució

Como vemos, al incorporar el tiempo siempre va a ocurrir que las cardinalidades cambian pasando de ser (0,1) o(O,n).

En el caso inicial, tenemos que un ejemplar sólo puede ser prestado a un socio. Esto quiere decir, que si otro socio coge el mismo ejemplar, perderemos la información relativa a quien cogió el ejemplar anteriormente. Es decir, la cardinalidad (O. 1) **no indica** que un ejemplar sólo va a ser cogido por un único socio 'para siempre' sino que ese ejemplar sólo puede relacionarse con un único socio a la vez.

e ullooooo

Análisis y diseño detallado de aplicaciones 1nformát,cas de gestión

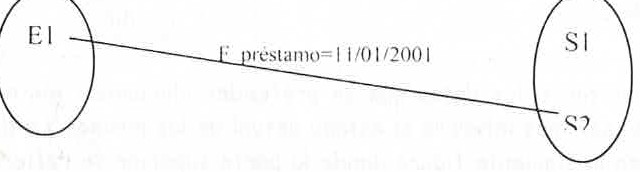
Veámoslo gráficamente:



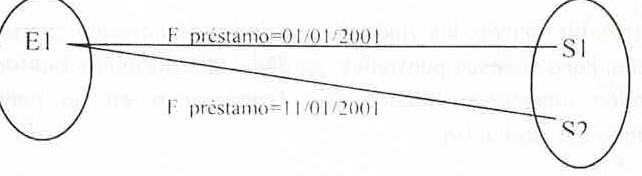
S2

F ,réswmo=O1/Ol /2001

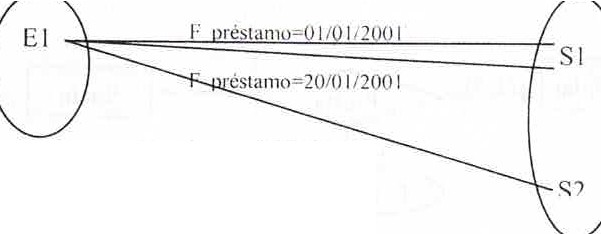
Ahora, si el ejemplar 1 (El) es devuelto por el socio S1 y posteriormente es cogido por el socio 52, lo información que quedará en nuestra Base de Datos será:



Perdiendo la información relativa a qué socio cogió el eJemplor El anteriormente. 51 pasamos al segundo ejemplo, en que lo card1nal1dad pasa o ser (O,n):



También puede suceder que el mismo ejemplar seo cogido por el mismo socio pero en fechas distintas:



F pr.:stamo= 11101,2001

Como se puede observar. el atributo que permite diferenciar que el mismo ejemplar es cogido por el mismo socio es el de F\_préstamo.

A veces resulta interesante representar la evolución de un tipo de entidad a lo largo del tiempo y aparece la noción de estado. Por ejemplo, si deseamos reflejar si un libro está en la biblioteca o se encuentro prestado, añadiremos al tipo de entidad un atributo que denominamos estado. que indicará en qué estado concreto se encuentra la entidad y que en muchos casos lleva asociado otro atributo, que es la fecha en la que se ha producido el cambio de estado; es también habitual en este tipo de aplicaciones que se desea tener constancia de la evolución de los estados, en cuyo caso se podría crear una nueva entidad, como SITUACIÓN, que tendría como atributos. entre otros posibles, estado y fecha.

Ejemplos:

* Colores.- Es el conjunto de los colores D*=* {rojo, azul, verde, etc.}
* Números enteros.- D ={O.± 1, ± 2, ± oc}
* Números\_de\_DNI.- Es el conjunto de números de DNI válidos, formados por ocho dígitos.

Edades\_de\_empleados .- Edades posibles de los empleados entre 16 y 80 años.

El producto cartesiano de un conjunto de dominios se define como:

* Dado un conjunto de dominios Dl, D2, ,... Dn (no necesariamente distintos), el producto cartesiano de

estos n conjuntos, denotado por Dl *x* D2 ***x* ...x** Dn, es el conjunto de todas las n tuplas ordenadas posibles <tl, t2, ..., tn> tales que cada ti e Di para i *=* 1..n.

***RELACIÓN*** Se puede definir relación como un subconjunto del producto cartesiano de una lista de dominios.

Def: Se define r como una relación sobre los conjuntos Dl, D2, ..., Dn si es un subconjunto del producto cartesiano Dl *x* D2 *x* ...x Dn. Siendo Dl, D2, ..., Dn. los dominios der.

De forma más sencilla se puede considerar una relación como una tabla de dos dimensiones, en la ue las co-lum-nas corresponden a los dominios y las filas contienen las tuplas. -

-

**A** cada columna de la tabla se le asigna un nombre. Así podemos definir :

Ai'rij.*r,"\,*

*ATRIBUTO* c o cada columna de la relación identificada por un nombre. Los atributos expresan las propiedades de los o jetos que nos interesan. Cada atri utoestará tomado de un dominio Di= dom (Ai). Varios atributos pueden tomarse del mismo dominio.

Hay que distinguir la diferencia entre dominio y atributo. Un atributo reeresenta el uso de un dominio

,,.\ MJ\(V,Q*<*

dentro de OQrelación. El dominio es un conjunto de valores del cual se sacan los valores que aparecen en una

f"'! *,t* "')

l

columna de la tabla. Normalmente se dan nombres distintos a los dominios y a los atributos para acentuar esta distinción.

***TUPLA*** es la lista ordenada den valores, t= { vl, v2,..., vn} donde cada valor vi, 1 i n, es un elemento de dom (Ai) o bien un valor nulo.

Una tupla corresponde a\_un conjunto de valores, tomados de las columnas de una tabla, para representar un objeto o un enlace entre objetos, del mundo real.

**3.3.**- ESQUEMA DE UNA RELACIÓN

Un esquema de una relación es el nombre de la relación seguido de la lista de !os atributos. Cada atributo estará tomado de un dominio.

Def: Un esquema de relación R, denotado por R (Al, A2,.... An), se compone de un nombre de relación, R, y de una lista de atributos, **A1, A2,** ..., **An.** Cada atributo Ai es el nombre del papel desempeñado por algún dominio D en el esquema R. Se dice que D es el dominio de Ai y se denota con dom **(Ai).**

El esquema de relación sirve para *describir* una relación **r.** El *grado de una relación* es el número de atributos, n, de su esquema de relación.

En toda relación hay que distinguir dos términos:

- INTENSIÓN de una relación.- Es el esquema R de la relación r, es decir, la definición de la relación que nos dice qué atributos la fo!:.\_man.

1. - EL MODELO RELACIONAL

**3.1.** - INTRODUCCIÓN

El modelo relacional de datos tuvo su origen en la publicación de los trabajos de E.F. cooo en 1970 (laboratorios IBM - Son José).

Se baso en uno estructuro de datos simple y uniforme - lo relación - y tienen fundamentos teóricos sólidos. El modelo relacional tiene asociado una teoría que nos p4ede ser separado del modelo: lo teoría de lo normalización de lo relaciones. Esto teoría tiene pgr objeto eliminar los comportamientos anormales de los relaciones durante los actualizaciones. También permite eliminar los datos redurdontes y facilito la comprensión de las relaciones semánticas entre los datos.

El modelo relacional es posterior a los modelos jerárquico y de red. Las desventajas de estos dos modelos condujeron a un intenso interés por el nuevo modelo de datos relacional. El modelo relocional do fueescrito or rimero z P-QC\_.eLDr. CODD en 1970, era un intento de simplificar la estructura de los Bases de Datos. Eliminaba las estructuras explícitas Padre/Hijo de la BD y en su lugar representaba todos los datos de la BD como sencillas tablas Fila/Columna de valores de datos; además argumentaba que los datos debían

relacionarse mediante interrelaciones lógicas. Estas tablas reciben el nombre de relaciones, de ahí el nombre del modelo.

Los primeros SGBD relacionales follaron en implementar algunas portes claves del modelo de cooo, que sólo ahora están encontrando acomodo en productos comerciales. Conforme el concepto relacional crecía en popularidad, muchas BD que se llamaban a sí mismos "relacionales" no lo eran de hecho, en respuesta a la corrupción del término *"re/acionaf,* el Dr. CODD, escribió un artículo en 1985 estableciendo 12 reglas a seguir por cualquier BD Relacional. Las 12 reglas de CODD han sido aceptadas desde entonces como la definición de un SGBD verdaderamente relacional, sin embargo, es más fácil la siguiente definición informal:

- Una Base de Datos Relacional es una BD en donde todos los datos visibles al usuario están organizados estrictamente como *tablas\_d es,* llamados *relaciones,* y en donde todas los operaciones de la BD operan sobre esas tabla

**3.2.-** CONCEPTOS BÁSICOS: DOMINIO, RELACIÓN, ATRIBUTO Y TUPLA.

La e ura básico y única del modelo relacional es la relación (también llamada\_ tabla), que sirve p\_ara representar tanto los objetos como las asociaciones entre ellos. Los atributos son los propiedades de los

relaciones, y se definen sobre los dominios. La existencia de un atributo va unido a la de la relación a la que pertenece.

El Universo del Discurso (es decir, aquella parte de la realidad de la cual queremos representar un modelo)

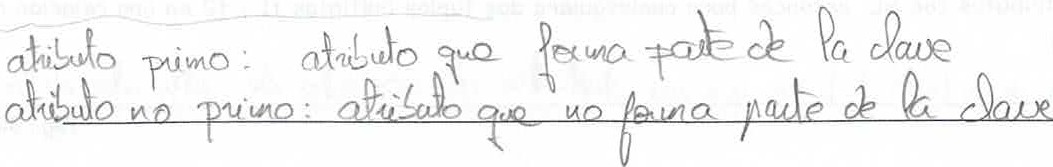
de una base de datos relacional está compuesto por un conjunto de dominios {D,}y de relaciones {R,} definidos sobre los dominios.

El concepto de relación deriva directamente de la teoría de conjuntos. Para dar una definición necesitamos en primer lugar introducir el concepto de dominio.

***DOMINIO.*** Conjunto finito y homogéneo de valores atómicos.

Por atómico se quiere decir que cada valor del dominio es indivisible en lo referente al modelo relacional.

Un método común en lo especificación de los dominios consiste en especificar el tipo de datos al cual pertenecen los valores del dominio, así como un nombre para el dominio que ayude o interpretar sus valores,y el formato de cada dominio, es decir, si son caract res, números enteros, etc. El dominio se pu de definir:\_pg\_r e\_:<tensión (dando sus posibles valores) o por intensión (mediante un tipo de dat s).

Todo conjunto de atributos SC de este tipo es una ***supere/ave*** del esquema de relación R. Toda relación tiene por lo menos una superclave: *el conjunto de todos sus atributos.* Sin embargo, una superclave puede tener atributos redundantes, as1 que un concepto más útil es el de ***clave,*** que carece de redundancia.

Una ***clave K*** de un esquema de relación R es una superclave de R con la propiedad adicional de que la eliminación de cualquier atributo A de K deja un conjunto de atributos K' que no es una superclove de R. Por tanto, una clave es una superclave mínima; una superclave a la cual no podemos quitarle atributos sin que deje de cumplirse la restricción deunicidad -

Ejemplo:

EMPLEADO (NSS, NOMBRE, EDAD, DIRECCIÓN, SALARIO)

El conjunto de atributos {NSS} es una clave de empleado porque no puede haber dos tuplas que tengan el mismo valor de NSS.

Cualquier conjunto de atributos que contengo a NSS, por ejemplo, {NSS, NOMBRE, EDAD} es una superclave, pero no uno clave porque si eliminamos NOMBRE o EDAD, o ambos del conjunto todavía tendremos una superclave.

La clave se determina a partir del significado de los atributos en el esquema de la relación, la propiedad no varía con el tiempo, debe seguir siendo válida aunque insertemos tuplas nuevas en lo relación. Por ejemplo, no podemos ni debemos designar como clave el atributo NOMBRE, porque no hoy garantías de que nunca existirán dos empleados con nombres iguales.

En general, un esquema de relación puede tener más de una clave. En tal caso, cada uno de ellos se denomina **CLAVE CANDIDATA.** Por ejemplo, la relación COCHE {Num\_matrículo, Num\_motor, Marca, Modelo, Año} tiene dos claves candidatas: Num\_matrícula y Num\_motor.

Es común designar a una de las claves candidatas como **CLAVE PRIMARIA** de la relación, ésta es la clave cuyos valores sirven para identificor las tuplas en una relación. Adoptaremos la convención de subrayar los atributos que forman la clave primaria de un esquema de relación.

Cuando un esquema tiene varias claves candidatas, la elección de una de ellas como clave primaria es arbitraria, sin embargo, casi siempre es mejor escoger una clave primaria con un solo atributo o un número reducido de atributos.

Una clave candidata que no es clave primaria recibe el nombre de **CLAVE ALTERNA.**

### 5. - RESTRICCIONES DEL MODELO RELACIONAL

Una base de datos relacional suele contener muchas relaciones y en éstas las tuplas están relacionadas de diversas maneras.

Un esquema de base de datos relacional Ses un *conjunto de esquemas de relaciones* S = {Rl, R2, ..., Rm} y un *conjunto de restricciones de integr1dadRI.* Un ejemplar de base de datos relacional BD de Ses un conjunto de ejemplares de relaciones BD= {r1 , r2, ..., rm} tal que cado r; es un ejemplar de Ri y tal que las relaciones r, satisfacen los restricciones de integridad especificadas en RI.

Las restricciones de integridad se especifican en el esquema de una base de datos y se deben cumplir en todos los ejemplares de ese esquema.

Entre los diversos tipos de\_c.es.tcicciones que se pueden especificar en un esquema de BD Relacional tenemos: las de dominio, dulave, de integridad de entidades y de integridad referencial.

* EXTENSIÓN de una relación.- Es la tabla con las tuplas, es decir, los valores en un momento dado.

Una relación se representa utilizando una tabla donde:

* Las columnas son los atributos que expresan las propiedades de la relación. El número de atributos se llama *grado de la relación.*
* Cada fila , llamada tupla, es un elemento del conjunto que es la relación. El número de tuplas se llama

*cardinalidad de la relación.* La cardinalidad varía con el transcurso del tiempo.

Ejemplo:

Nombre de la relación Atributos

>- **INTENSIÓN**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| COCHE  1 | NMATRIC | 1 | MARCA | 1 | TIPO | 1 POTENCIA | 1 | COLOR |
| /' | M 8419 EN | RENAULT | | R 12 TS | | 6 | ROJO | |
| .........  Tuplas | B 1411 AX  BA 0014 AE | PEUGEOT CITROl:N | | 504  2 *CV* | | 9  2 | VERDE AZUL | |

>- **ÉXTENSIÓN**

Una vez definidos los conceptos de dominio, atributo y tupla podemos dar otra definición de relación:

Def: Una relación r de esquema de relación R (Al, A2, ... , An) denotado también por r (R), es un conjunto de n tuplas r *=* {tl, t2,... , tn}. Cada tupla tes una lista ordenada den valores, t ={vl, v2, ... , vn} donde cada valor vi, 15 i *5* n, es un elemento de dom (Ai) o bien un valor nulo.

Podemos visualizar una relación como una tabla de valores, cada fila de la tabla representa una colección de valores de datos relacionados entre sí, que describen una entidad del mundo real. El nombre de la tabla y los nombres de las columnas ayudan a interpretar el significado de los valores que están en cada fila de la tabla.

En terminología del modelo relacional, una fila se denomina *tupla,* una cabecera de columna es un *atributo* y la tabla es una *relación.* El tipo de datos que describe los tipos de valores que pueden aparecer en cada columna se llama *dominio.*

En general, a medida que cambia el estado del mundo real, cambia la relación.·Sin embargo, el esquema de la relación es relativamente estático y no cambia con frecuencia, lo hace cuando por ejemplo se añade un atributo para representar información nueva que originalmente no estaba en )a relación.

3.4. - CLAVES DE LAS RELACIONES

Una clave es un atributo o conjunto de atributos cuyos valores son únicos dentro de una relación y por lo tanto se pueden usar para identif icor de forma única a cada tupla de todas las demás de una relación.

Una relación se define como un conjunto de·tuplas. Por definición, todos los elementos de un conjunto son distintos; por tanto, todas las tuplas de una relación deben ser distintas. Esto significa que no puede haber dos tuplas que tengan la misma combinación de valores para *todos* sus atributos. Por lo general, existen subconjuntos de atributos de un esquema de relación R con la propiedad de que no debe haber dos tuplas en una relación r de R con la misma combinación de valores para estos atributos. Supongamos que denotamos un

subconjunto así de atributos con SC; entonces para cualesquiera dos tuplas distintas tl y t2 en una relación r de R, tl(SC)*\** t2(SC).

Caju.J-1:> e\: alub.k ck º°' *t fc. l* b r cL.1ttJo *J* ro to c;)q J J *ceOaL L.*

Un conjunto de atributos CE en el esquema de relación Rl es una **clave externa** de Rl si satisface las dos reglas siguientes:

1.- Los atributos de CE tienen el mismo dominio que los atributos de la clave primaria CP de otro esquema de relación R2; se dice que los atributos CE **hacen referencia** o **se refieren** a la relación R2.

2.- Un valor de CE en una tupla tl de Rl ocurre como valor de CP en alguna tupla t2 de R2 o bien es nulo. En el primer caso, tenemos tl(CE]= t2(CP], y decimos que la tupla tl **hace referencia** o **se refiere** a la tupla t2.

Por ejemplo supongamos las relaciones: EMPLEADO que posee entre sus atributos Ndep (nº de departamento para el cual trabaja el empleado) y DEPARTAMENTO con uri atributo NumD (indica el nº del departamento en la compañía). Designamos a Ndep como clave exte;na·de EMPLEADO, con referenciaa la relación DEPARTAMENTO. Esto significa que un valor de Ndep en cualquier tupla tl de la relación EMPLEADO deberá coincidir con un valor de la clave primaria de DEPARTAMENTO - el atributo NumD - en alguna tupla t2

de la relación DEPARTAMENTO, o el valor de Ndep puede ser nulo si el empleado no pertenece a ningún departamento.

En una BD con muchas relaciones, suele haber muchas restricciones de integridad referencial. Las restricciones de integridad referencial surgen casi siempre de los vínculos entre las entidades representadas por los esquemas de relaciones.

Cabe señalar que una clave externa puede hacer referencia a su *propia relación.* Por ejemplo, el atributo NSSSUPER de EMPLEADO se refiere al supervisor de un empleado, el cual es otro empleado representado por una tupla de la relación EMPLEADO. Así pues, NSSSUPER es una clave externa que hace referencia a la relación EMPLEADO misma.

Podemos representar gráficamente las restricciones de integridad referencial trazando un arco dirigido de cada clave externa a la relación a la cual hace referencia.

Debemos especif icor todas las restricciones de integridad en el esquema de la BD relacional si es que nos interesa mantener dichas restricciones para todos los ejemplares de la BD. En consecuencia, en un sistema relacional, el lenguaje de definición de datos (DDL) debe contar con mecanismos para especificar los diversos tipos de restricciones para que el SGBD pueda imponerlas automáticamente.

Los tipos de restricciones que hemos visto no incluyen una amplia clase de restricciones generales, a veces llamadas *restricciones de integridad semántica,* que puede ser necesario especificar e imponer en una BD relacional. Ejemplos de tales restricciones son "el salario de un empleado no debe exceder el salario de su supervisor", "el número máximo de horas que un empleado puede trabajar en todos los proyectos por semana es 56", etc.

### 6. - DISEÑO DE BASES DE DATOS RELACIONALES

En general, el objetivo del diseño de una base de datos relacional ees nerar un conjunto de esquemas de relaciones que permitan almacenar la informaci<?.!c!.o.n un mínimo de redundanci , pero que a la vez faciliten la recuperacióñae la información. Una de las técnicas para lograrlo consiste en diseñar esquemas que tengan una *forma normal* adecuada. Para determinar si un esquema de relaciones tiene una de las formas normales se requiere mayor información sobre la empresa del "mundo real" que se intenta modelar con la base de datos. La información adicional la proporciona una serie de limitantes que se denominan *dependencias de los datos.*

#### - Restricciones de dominio.

Las restricciones de dominio especifican que el valor de cada atributo A deber ser un valor atómico del dominio dom(A) para ese atributo. Los tipos de datos asociados a los dominios por lo regular incluyen los tipos de datos numéricos estándar de los números enteros (entero-corto, entero, entero-largo) y reales (flotante y flotante de doble precisión). También disponemos de caracteres, cadenas de longitud fija y cadenas de longitud variable, así como tipos de datos de fecha, hora, moneda. Otros dominios posibles se pueden describir mediante un subintervalo de valores de un tipo de datos o como un tipo de datos enumerado en el que se listan explícitamente todos los valores posibles.

#### - Restricciones de clave.

Una relación se define como un conjunto de tuplas. Por definición, todos los elementos de un conjunto son distintos; por tanto, todas las tuplas de una relación deben ser distintas. Esto significa que no puede haber dos tuplas que tengan la misma combinación de valores para *todos* sus atributos. Por lo general, existen otros subconjuntos de atributos de un esquema de relación R con la propiedad de que no debe haber dos tuplas en una relación r de R con la misma combinación de valores para estos atributos. Supongamos que denotamos un subconjunto así de atributos con SC; entonces para cualesquiera dos tuplas distintas tl *y* t2 en una relación r

de R, tenemos la siguiente restricción: tl[SC]*\** t2[SC].

Todo conjunto de atributos SC de este tipo es una ***supere/ave*** del esquema de relación R. Toda relación tiene por lo menos una superclave: *el conjunto de todos sus atributos.* Sin embargo, interesa la superclave mínima **(clave)** una superclave a la cual no podemos quitarle atributos sin que deje de cumplirse la restricción de unicidad.

El valor de un atributo clave puede servir para identif icor de manera única una tupla de la relación. El hecho de que un conjunto de atributos constituya una clave es una propiedad del esquema de la relación; es una restricción que debe cumplirse en *todos* los ejemplares de relaciones del esquema; por tanto, la propiedad *no varía con el tiempo;* debe seguir siendo válida aunque insertemos tuplas nuevas en la relación.

Recordamos que hay distintos tipos de claves, ya comentados.

#### - Restricción de integridad de entidades.

La restricción de integridad de entidades establece que nin ún valor de clave rimaria puede ser nulo. Esto es porque el valor de la clave primaria sirve para identificar las tuplas individuales de una relación; el que la clave primaria tenga valores nulos implica que no podemos identificar algunas tuplas. Por ejemplo, si dos o más tuplas tuvieran nulo en su clave primaria, tal vez no podríamos distinguirlas.

Las restricciones de clave y de integridad de entidades se especifican sobre relaciones individuales.

#### - Restricción de integridad referencial.

La restricción de integridad referencial se especifica entre rfas relaciones y. sirve paca mnn;teo.ec. la

consistencia entre tuplas de las dos relaciones. En términos informales, la restricción de integridad referencial establece que una tupla en una relación que haga referencia a otra relación deberá referirse a una *tupla existente* en esa relación.

Para dar una definición más formal de integridad referencial primero debemos definir el concepto de **clave externa.** Las condiciones que debe satisfacer una clave externa especifican una restricción de integridad referencial entre los dos esquemas de relaciones Rl y R2.

**3.6.2.- DEPENDENCIAS FUNCIONALES** *(* J? ***,e (Ql.Qv\. de \o-* /J** VIO

le. e<J .s. '-")

***Concepto de dependencia funcion*** / \_ \ I:'\ • . 1 **L** \ *{\* '

""'- do *x->A* a):'li o *x tP ca* G. cG-V'IS'(l ei1. 0-fu u.ro A JQ 1'0'11 c..:a'1

l

Codd introdujo el concepto de dependencia funcional para caracterizar aquellas relaciones que pueden descomponerse sin pérdida de informaciones.

t -i,J:r

# . . . ¼,kl(l(')-Yt?(}

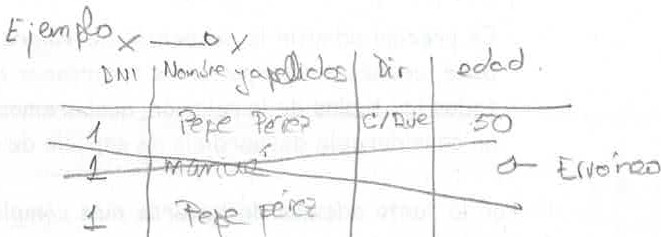
- Dependencia Funcional. Sea R (A1, A2, ..., An) un esquema de relación, y X e Y dos subconjuntos de { A1, A2, .,..

An}, Se dice que X Y (X determina a Y, o que Y depende funcionalmente de X) si para toda extensión r de R, para toda tupla t1y t2 de r, se tiene: •

7l X (t1) = 7l X (t2) 7l Y (ti) = 7l y (t2)

Más sencillo: un atributo (o grupo de atributos ) Y depende funcionalmente de un atributo (o grupo de atributos) X si, a un valor dado de X le corresponde un único valor de Y (y sea cual sea el instante considerado).

O también X determina funcionalmente a Y en un esquema de relación R si y sólo si, siempre que dos tuplas coincidan en su valor X, necesariamente deben coincidir en su valor Y.

Ej: Si tenemos R = COCHE= **{NM, MARCA,** TIPO, POTENCIA, COLOR} encontramos las siguientes dependencias funcionales:

**NM** COLOR TIPO **MARCA** TIPO POTENCIA

(TIPO, **MARCA)** POTENCIA

pero no existen

POTENCIA TIPO TIPO COLOR

Siempre se verifica que:

* **X** 0 (el conjunto vacío depende de cualquier conjunto de atributos).
* hay una dependencia funcional siempre de las claves, por ejemplo si tenemos R= { DNI, **A1, A2,** ..., An} cualquier **A;** depende de DNI (DNI **A;).**

Las dependencias funcionales se extraen de la semántica de los atributos (significado de los atributos, es decir, de la relación que hay entre ellos), y deben cumplirse para *todas* las posibles extensiones r de R Siempre que la semántica de dos conjuntos de atributos de R indique que debe cumplirse una dependencia funcional, se especificará esta dependencia como una *restricción.*

Una dependencia funcional es *una\_p.ropiedad del esquema de relación* (intensión) R, *y* no de un ejemplar de relaci6n (extensión) r de R en particular. Por ello, una DF *no puede* inferirse automáticamente de una extensión de relación *r* dada, sino que debe definirla explícitamente alguien que conozca la semántica de los atributos.

Las dependencias funcionales se usan para formular el conocimiento que tenemos sobre los datos, buscando la minimización de la redundancia *y* mejoras en disponibilidad y uso.

***Propiedades de las dependencias funciona/es***

Las dependencias funcionales cumplen varias reglas:

* + 1. *Reflexividad:* Y \;; **X X** Y, todo conjunto de atributos determina a sí mismo o a una parte de sí m;,m, t o d l:O..'f''€\_ *J t* l=W:\_ *EUJ,*

v

*Qs*/""\

*:*

# ) .1fj

)

Análisis y diseño detallado de aplicaciones informáticas de gestión

### 3.6.1.- PROBLEMAS EN EL DISEÑO DE BASES DE DATOS RELACIONALES

'

*li p*

Antes de hablar de formas normales y dependencias de datos es conveniente considerar los defectos que

1 1 puede tener una base de datos mal diseñada.

*'4* Supongamos las siguientes relaciones:

Q PERSONA (DNI, NOMBRE, APELLIDOS)

...

*J*

COCHE **(NM. MARCA,** TIPO, POTENCIA, COLOR) TENER (DNI, **NM,** FECHA, PRECIO)

\.. t J -

Si en lugar de las anteriores relaciones que componen lo BD, optásemos por una única relación, formada por los atributos de las tres, ésta tendría los siguientes defectos:

* En primer lugar, algunos datos serán ***redundantes.*** en general en esta relación 4na persona aparecerá tantas veces como coches posea.
* Esta redundancia conlleva unos ***riesgos de incoherencia*** durante las actualizaciones; por ejemplo, si

j resulta que el nombre de López no es Pedro sino Juan, hay que tener cuidado y actualizar +odas las tuplas

1J

\_,4---

*i.*

en las que aparece López.

- Es preciso admitir la presencia de ***valores*** *nulo* en una relación de este tipo para poder mantener en la base, coches sin propietarios o personas que no tienen coches. Si muchos de los atributos no se aplican a todas las tuplas de la relación, acabaremos con un gran número de nulos en esas tuplas. Esto puede originar un considerable desperdicio de espacio de almacenamiento.

*l...\_)* j P lo ademá.s de hacerse más complicada la actualización (inserción, eliminación y modificación),\_g

'(} *--9.b*í desperdicia espacio. o \ \\.

l

<,¡ , - *r.¡j*

Uno de los objetivos en el diseño de esquemas es *minimizar el espacio de almacenamiento* que ocupan las

f 1 :." relaciones base (archivos). La agrupación de atributos en esquemas de relación tiene un efecto significativo

-.,.

.¡ *ó*

t..f::., sobre el espacio de almacenamiento, se requiere más. •

~J •

rO h s- "ti

1----i ¼-1 *Aproximación por descomposición.*

La aproximación por descomposición para oncebir esquemas relacionales parte de una relación compuesta

de todos los atributos, llamada re *ación universal ara* descomponer después esta relación en subrelaciones que no padecen las anomalías anteriormente citadas. El proceso es un proceso de depuración sucesiva que debe

lograr aislar unas entidades, y 4-nas aso\=ia,'rion s del mundo real. O - 1°dv..,O::- d **Cl)vq &tqi**

0,t \_

C.i:.c"' et:>"' e"b11Ocá. ::'.:ti.:. r:Voú. . { *\_1',* <qJ" ·i,< " ;--re.<o U/lo. "\_x,r *cbcJ* re ni\ 00

'n*'ef•C:'if* La teoría de la descomposición de las relaciones utiliza dos operaciones fundamentales de manipulación de

G relaciones: *la proyecc16n y la reunión natural.* .

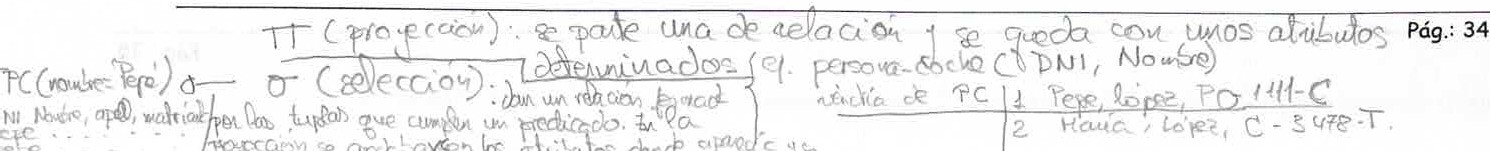
La ***descomposición*** consiste en la sustitución de la relación R **(A1, A2,** ... , An) por una serie de relaciones Rl, R2, ..., Rn obtenidas mediante proyecciones de R y tales que la relación resultante de las reuniones Rl D<J R2 D<l ... D<l Rn tiene el mismo esquema que R.

Pero a veces las descomposiciones elegidas de R no permiten re-cuperar algún valor; por lo que debemos hacer ***descomposiciones sin pérdidas*** que será la descomposición de una relación R en Rl, R2, ..., Rn tal que para toda extensión de R se tiene: R=Rl 1XJ R2 D<J ... CXl Rn.

El problema de concebir unas bases de datos relacionales puede asimilarse al de descomponer la relación

universal compuesta de todos los atributos en subrelaciones que no sufrat:i l s omalías vis\as anteriormente y obtener una descomposición sin pérdida.(\_VIO a:, ce i 1,C'c)QI- w,.' al,u.!., 05 w.. c,t)

tf'



La ***normalización de los datos*** puede considerarse como un proceso durante el cual los esquemas de relación insatisfactorios se descomponen repartiendo sus atributos entre esquemas de relación más pequeños que

poseen propiedades deseables. Un objetivo del proceso de normalización es garantizar que no ocurran anomalías de actualización.

Las formas normales, consideradas aparte de otros factores, no garantizan un buen dis ño de BD. En general, no basta con comprobar por separado que cada esquema de relación de la BD esté en, digamos, FNBC o 3FN. Más bien, el proceso de normalización por descomposición debe confirmar la existencia de propiedades adicionales que los esquemas relacionales, en conjunto, deben poseer. Dos de estas propiedades son:

* La propiedad de reunión sin pérdjdp, que garantiza que no se presentará el problema de las tuplas erróneas.
* La propiedad de conservación de las dependencias, que asegura que todas las dependencias funcionales estén representadas en algunas de las relaciones individuales resultantes.

La utilidad práctica de las formas normales queda en entredicho cuando las restricciones en las que se basan son difíciles de entender o de detectar por parte de los diseñadores de BD y usuarios que deben descubrir estas restricciones.

Otro punto que merece la pena destacar es que los diseñadores de BD *no tienen que* normalizar hasta la forma normal más alta posible. Las relaciones pueden dejarse en formas normales inferiores por razones de rendimiento (Ej: la relación EMP-DEPTO (NOMBRE, NSS, FECHAN, DIRECCIÓN, NÚMEROD, NOMBRED,

NSSGTED) la dejaríamos así, sin dividir, si por ejemplo una consulta importante obtiene información relativa al departamento de un empleado, junto con los atributos del empleado. Pero hay que tomar nota de sus anomalías y entenderlas perfectamente de modo que, al actualizar la relación, no se produzcan inconsistencias).

#### Primera Forma Normal (1FN)

Una relación está en **primera forma normal** si todos los atributos contienen v or s atómicos (e el sentido, de tomados de un único dominio y que tienen un único valor). \.Jo \t..c,. .'t\-- *a.11.u.aUCG* \U

cl)'At tr.>( *..ó"°' O)*

La primera forma normal se definió para prohibir los atributos multivaluados, compuestos y sus combinaciones.

Cuando una relación no está en primera forma normal, se divide en otras relaciones, repartiendo sus atributos entre las resultantes. Normalmente la idea es eliminar el atributo que viola la lFN de la relación original y colocarlo en una relación aparte junto con la clave primaria de la relación de partida.

*Segunda forma normal (2FN)*

La segunda forma normal se basa en el concepto de ependencia funcional total. Una dependencia funcional

X Y es una *dependencia funcional total* si la eliminación de cualquier atributo A de X hace que la dependencia deje de se válida, en caso contrario se\_r,á una *dependencia funcion arcial* . . .

*1*

Un atributo del esquema de relac1on R, se denomina *atr,but primo* de R s1 es miembro de cualquier clave candidata de R. si no es miembro de ninguna clave candidata se I denominará *atributo no primo.*

Una relación está en segunda forma normal si:

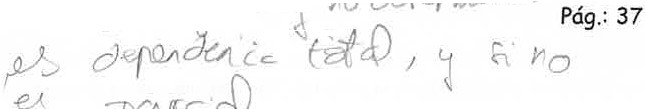
* está en primera forma normal *(*
* y todo atributo no primo **A** en R depende funcionalmente de manera tot I de cada clave candidata de R.

. .otda:> -o *c1 e I*

c..o11

### \\)o"''fi-\ ,

:IVI



\

*:* L,,

uJc 1

lr CG.t:

o Jg,fep,,tl*o*

 Análisis y diseño detallado de aplicaciones informáticas de gestión

*i)*

*11 arr k J* of(i.:;:"(D..f'll'{lqc CQ..

*"•*

* + 1. *Aumento:* X Y :=:) ZX ZY; si X determina a Y, los dos conjuntos enriquecidos por un tercer conjtmto. L ll O
    2. *Transitividad:* X Y e Y Z X z -e¡·Q,r-f *<J* Nvir -" u t'- L -•

de atributos pueden verse

lr') e. t:pc - F°b(.,, e, =)

Nu...,- ,,/*r, De.* l° i *e* e.e.

Las tres reglas anteriores se conocen como *Axiomas de Armstrong.* A partir de estos tres axiomas se deducen otras reglas más:

* + 1. *Aditividad (o unión):* X Y y X Z X YZ
    2. *Proyectiv1dad (o descomposición):* X YZ X Y y X Z (si un conjunto de atributos depende de otro, cada uno de ellos depende de este otro).
    3. *Pseudo-transitividad:*X Y y WY Z WX Z

A partir de las reglas anteriores , se puede introducir el concepto de dependencia funcional elemental de la forma **X A,** donde **A** es un atributo único no incluido en **X (A** *cz.* **X)** y donde no existe un X' e X tal que **X' A.**

Si tenemos un conjunto de DF elementales F, llamamos ***cierre·***del conjunto de dependencias funcionales denotado por F', a todas las dependencias funcionales que podamos obtener aplicando a F los axiomas anteriores.

Dos conjuntos de dependencias funcionales son E y F son *equivalentes* si sus cierres coinciden , E' = F'.

#### Claves de una relación

Debemos distinguir entre *supere/ave* y *clave.* Una **superclave** de un esquema de relación R **(A1, A2,** ... , An) es un conjunto de atributos 5 **R** con la propiedad de que no habrá un par de tuplas t1y t2 en la extensión r de R

tal que t1 (5=) t2 (5). Una **clave** K es una superclave con la propiedad adicional de que la eliminación de cualquier

atributo de K hará que K deje de ser una superclave. La diferencia entre .una clave y una superclave es que la primera tiene que ser *"mínima''.*

Puede haber varias claves mIrnmas para una misma relación *(claves candidatas),* pero se elige arbitrariamente una de ellas como *clave primaria* y las demás serán *claves secundarias.*

Ahora daremos una definición de clave a partir de la de dependencia funcional.

- **Clave de una relación:** Subconjunto X de los atributos de una relación R **(A1, A2,** ... , An) tal que:

* + - 1. - X A1 A2 ... An
      2. - No existe ningún conjunto Y e X tal que Y AI A2 ... An

En otras palabras, una clave es un *conjunto mínimo de atributos qu determina a todos los demás.*

3.6.3.- NORMALIZACIÓN DE RELACIONES ( *rNJe(rt.JaJ* ,"pru,r,.ac o ¡;,:V1... *-pe-r&:d )*

En el proceso de normalización, según la propuesta original de Codd (1972), se somete un esquema de relación a una serie de pruebas para "certificar" si pertenece o no a una cierta *forma normal.* En un principio, Codd propuso tres formas normales, a las cuales llamó primera, segunda y tercera formas normales (lFN, 2FN, 3FN). Posteriormente, Boyce y Codd propusieron una definición más estricta de 3FN, a la que se conoce como forma normal de Boyce-Codd (FNBC). Todas estas formas normales se basan en las dependencias funcionales entre los atributos de una relación. Más adelante se propusieron una cuarta forma normal (4FN) y una quinta

(5FN), con fundamento en los conceptos de dependencias multivaluadas y dependencias de reunión, respectivamente.

### 3.7.1. - ALGORITMO DE TRANSFORMACIÓN E/R-RELACIONAL

Describiremos informalmente los pasos de un algoritmo para la transformación del modelo E/R Básico al modelo Relacional.

***Paso J:*** Por cada tipo normal de entidades *E* del esquema ER, se crea una relación *R* que contenga todos los atributos simples de E. Se incluyen sólo los atributos simples componentes de un atrjbqto eomr.ue20. Se elige uno de los atributos clave de E como clave primaria de *R.* Si la clave elegida, es compuesta, el conjunto de atributos simples que la forman constituirá la clave primaria de *R.*

***Paso 2:*** Por cada tipo de entidad débil *D* del esquema ER con tipo de entidades propietarias *E,* se crea una relación *R* y se incluyen todos los atributos simpl s (o componentes simples de los atributos compuestos) de *D* como atributos de *R.* Además, se incluyen como atributos de clave externa de *R* los atributos de clave primaria de la relación o relaciones que corresponden al tipo o tipos de entidades propietarias; con esto damos cuenta del tipo de vínculo identificador de *D.* La clave primaria de Res la combinación de las claves primarias de las propietarias y la clave parcial de *D,* si existe.

***Paso 3'.*** Por cada tipo de vínculo binario 1:1 *R* del esquema ER, se identifican las relaciones *5* y *T* que corresponden a los tipos de entidades que participan en *R.* Se escoge una de las relaciones -digamos *5-* y se incluye como clave externa en *5* la clave primaria de *r* Es mejor elegir un tipo de entidades

con participación total en *R* en el papel de *5.* Se incluyen todos los atributos simples (o componentes simples de los atributos compuestos) del tipo de vínculos 1:1 *R* como atributos de *5.*

**NOTA:** Cabe señalar que puede establecerse una transformación alternativa de un tipo de vínculos 1:1 si combinamos los dos tipos de entidades y el vínculo en una sola relación. Esto resulta apropiado sobre todo cuando las dos participaciones son totales y cuando los tipos de entidades no participan en ningún otro tipo de vínculos.

*Paso* ***4:*** Por cada tipo de vínculo normal (no débil) binario 1:N *R,* se identifica la relación *5* que representa el tipo de entidades participante del *lado N* del tipo de vínculos. Se incluye como clave externa en *5* la clave primaria de la relación *T* que representa al otro tipo de entidades que participa en R; la razón es que cada ejemplar de entidad del lado *N* está relacionado con un máximo de un ejemplar de entidad del lado 1. Se incluyen todos los atributos simples (o componentes simples de los atributos compuestos) del tipo de vínculos 1:N como atributos de *5.*

*Paso 5:* Por cada tipo de vínculo binario *M:N R,* se crea una nueva relación *5* para representar *R.* Se incluyen como atributos de clave externa en *5* las claves primarias de las relaciones que representan los tipos de entidades participantes; su combinación constituirá la clave primaria de *5.* También se incluyen todos los atributos simples (o componentes simples de los atributos compuestos) del tipo de vínculos **M:N** como atributos de *5.* Hay que observar que no podemos representar un tipo de vínculos **M:N** con un solo atributo de clave externa en una de las relaciones participantes -como hicimos en el caso de los tipos de vínculos 1: 1 y 1:N debido a la razón de cardinalidad **M:N.**

**NOTA:** Cabe destacar que siempre es posible transformar los vínculos 1: 1 o 1:N de una manera similar a como se hace con los vínculos M:N. Esta alternativa es útil sobre todo cuando hay pocos ejemplares del vínculo, a fin de evitar valores nulos en las claves externas. En este caso, la clave primaria de la relación "vínculo" será la clave externa de sólo una de las relaciones "entidad" participantes. En el caso de un vínculo 1:N, ésta será la relación entidad del lado N; en el caso de un vínculo 1:1, se elegirá la relación entidad con participación total (si existe).

***Paso 6:*** Por cada atributo multivaluado ***A*** se crea una nueva relación *R* que contiene un atributo correspondiente a *A* más el atributo de clave primaria K (como clave externa en *R)* de la relación que representa el tipo de entidades o de vínculos que tiene a *A* como atributo. La clave primaria de Res la combinación de *A* y *K* Si el atributo multivaluado es compuesto, se incluyen sus componentes simples.

Análisis y diseño detallado de aplicaciones informáticas de gestión

Si un esquema de relación no está en 2FN, se le puede normalizar a varias relaciones en 2 FN en las que los atributos no primos estén asociados sólo a la parte de la clave primaria de la que dependen funcionalmente de manera total.

*Tercera forma normal {3FN)*

La tercera forma normal se basa en el concepto de dependencia transitiva. Una dependencia funcional X Y en un esquema de relación R es una *dependencia transitiva* si existe un conjunto de atributos Z que no sea un subconjunto de cualquier clave de R, y se cumplen tanto X Z como Z Y.

Una relación está en tercera forma normal si:

* está en 2FN
* y todo atributo no primo de R no depende transitivamente de alguna clave candidata, es decir, depende de manera no transitiva de toda clave candidata.

Otra definición de 3FN sería: Un esquema de relación R está en 3 FN si, siempre que una dependencia funcional X A se cumple en R, o bien (a) X es una superclave de R, o (b) A es un atributo primo de R. Un

esquema de relación R viola esta definición de 3FN si una dependencia funcional X A válida en R viola ambas condiciones, (a) y (b).

*Forma normal de Boyce-Codd (FNBC)*

-=-

La forma normal de Boyce-Codd es más estricta que la 3FN, lo que significa que toda relación que esté en FNBC también está en 3FN; sin embargo, una relación en 3FN *no estánecesariamente* en FNBC.

La definición de FNBC difiere un poco de la definición de 3 FN.

Un esquema de relación R está en forma normal de Boyce-Codd (FNBC) si, siempre que una dependencia funcional X A es válida en R, entonces X es una superclave de R.

La única diferencia entre FNBC y 3FN es que la condición de la 3FN que permite que **A** sea un atributo primo de R si X no es una superclave, está ausente en la FNBC.

En la práctica, casi todos los esquemas de relación que están en 3FN también estcin en FNBC. Sólo si existe

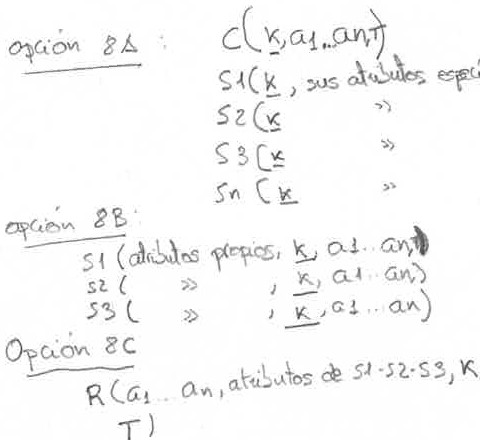
una dependencia X A en un esquema de relación R, y X no es una superclave y **A** es un atributo, Restará en 3FN pero no en FNBC.

Es mejor tener los esquemas de relación en FNBC; si esto no es posible bastará con que estén en 3FN. Sin embargo. ni 2FN ni 3FN se consideran buenos diseños de esquemas de relación. Estas formas normales se desarrollaron históricamente como escalones para llegar a 3FN y FNBC.

A veces las descomposiciones en FNBC no conservan todas las DF, entonces si el diseñador se ve obligado a escoger entre la FNBC o conservar las dependencias funcionales con la 3FN. se opta por ésta última (que es más débil). Pero si el diseño en FNBC mantiene todas las DF se opta por FNBC.

## 3. 7. - REGLAS DE PASO DEL MODELO E/R AL MODELO RELACIONAL

Aquí explicaremos la forma de derivar un esquema de BD Relacional a partir de un esquema conceptual creado empleando el modelo E/R.



***Paso*** 7: Por cada tipo de vínculo n-ario *R,* n > 2, se crea una nueva relación *5* que represente a *R.* Se incluyen como atributos de clave externa en *5* las claves primaria de las relaciones que representan los tipos de entidades participantes. Tambiéri se incluyen los atributos simples (o los componentes simples de los atributos compuestos) del tipo de vínculos n-ario como atributos de *5.* La clave primaria de *5* casi siempre es una combinación de todas las claves externas que hacen referencia a las relaciones que

representan los tipos de entidades participantes. No obstante, si la restricción de participación *(mín, máx)* de uno de los tipos de entidades *E* que participan en R tiene *máx*= 1, la clave primaria de S podrá ser el único atributo de clave externa que haga referencia a la relación *E'* que corresponde a *E;* la razón es que, en este caso, cada uno de las entidades *e* de *E* participará en cuando más un ejemplar de vínculo de *R* y, por tanto, podrá identificar de manera único ese ejemplar.

3.7. 2.- TRANSFORMACIÓN EER-RELACIONAL

Ahora analizaremos la transformación de los conceptos del modelo E/R Extendido (EER). para ello podemos añadir un poso más o nuestro algoritmo de transformación ER-Relocionol, que tiene 7 pasos, para manejar la transformación de la especialización. Usaremos Atrs(R) poro denotar los atributos de lo relación r y CLP(R) paro denotar lo clave primaria de R.

***Paso 8.*** Convertir cada especialización con m sG'tr es {51, 52,... ,Sm,} *y* s'üpe se (generalizada) C, donde los atributos de C son {k, 01,... , on} *y* k es lo clave (primario), en esquemas de relación empleando una de estos cuatro opciones:

***Opción 8A:*** Crear una relación para la superclase *y* tontos relaciones como subclases hayo. Creamos

una relación L para C con atributos Atrs (L=) {k, 01,... ,an,} y CLP(L) = k. Crear uno relación L,,. por coda

subclase S;, l<= i <= m, con los atributos Atrs(L;) = {k} v {atributos de S;} y CLP(L;) = k.

) ***O ción 88.*** Crear uno relación por codo subclase, que contenga, además de los atributos propios, los

. atributos comunes. Creamos una relación L; por cado subclase Si, l<= i <= m, con los atributos Atrs(L;)

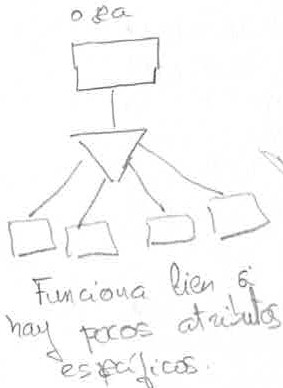
}- = {atributos de S;} v {k, 01,...,on,} y CLP(L;) = k.

'l'\O *\&c..o"'-0-* -

(1• e>;. (.,l. ***Opción 8C.*** Crear una solo relación con los atributos de la superclose y de los subclases. Creamos una

U!'- (L:,;, c;.. / sola relación L con atributos Atrs(L)= {k, 01,..., On} v {atributos de 51} v ... v {atributos de Sm} v {t} Y

\ *V\O JI* CLP(L) = k. Esta opción es poro una especialización cuyas subclases son disjuntas, y t es un atributo

*J* v de tipo que indico la subclase a la que pertenece cado tupla, si la hay. Esta opción tiene el potencial e crear un gran número de valores nulos.

*Opción* ***8D:*** Crear un solo esquema de relación L con atributos Atrs(L)*=* {k, 01, ... , On} v {atributos de

51} v ... v {atributos de Sm} v {t1, t2,..., tm} *y* CLP(L)= k. Esto opción es paro uno especialización cuyos subclases no son disjuntas *y* coda t;, 1 <= i <= m, es un atributo booleano que indico si una tupla

"' pertenece o no a una subclase S;.



...