

### UNIDAD 3: EL MODELO RELACIONAL

El modelo de datos relacional fue introducido por Codd (1970). Se basa en una estructura de datos simple y uniforme –la relación- y tiene fundamentos teóricos sólidos. El modelo relacional se ha establecido firmemente en el mundo de las aplicaciones de base de datos, y existen en el mercado muchos paquetes de SGBD relacionales. Se puede decir que hoy en día, es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Es un modelo de datos basado en la lógica de predicado y en la teoría de conjuntos.

El modelo relacional representa la base de datos como una colección de relaciones. En términos informales, cada relación semeja una tabla. Si visualizamos una relación como una tabla de valores, cada fila de la tabla representa una colección de valores de datos relacionados entre sí. Dichos valores se pueden interpretar como hechos que describen una entidad o un vínculo entre entidades 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.

Por ejemplo, la tabla de la Figura 1 se llama LIBROS porque cada fila representa hechos dados acerca de una entidad libro en particular. Los nombres de las columnas –ISBN, Nombre, Genero, Estado, Sig\_Top, Nro\_I, Editorial, Año\_de\_Ed, Cant\_pag – especifican como interpretar los valores de datos de cada fila, con base en la columna en que se encuentra cada valor. Todos los valores de una columna tiene el mismo tipo de datos.

En terminología del modelo relacional, una fila se denomina *tupla*, una cabecera de columna *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*. A continuación definiremos estos términos –*tupla*, *atributo*, *dominio*, *relación*- con mayor precisión.

ISBN	Nombre	Genero	Sig_Top	Nro_I	Editorial	Año	pag
0-13-861337-0	A First Course in Database System	Técnico	A-12-456	827	Prentice Hall	1997	470
950-731-260-9	La Resistencia	Novela	C-45-896	956	Planeta	2000	148
950-9122-55-6	La Maquina Cultural	Novela	C-78-965	124	Planeta	1998	292
0-415-11966-9	Wholeness and the Implicate Order	Novela	D-45-655	456	Routledge	1980	224
950-13-6183-7	Apuntes y Aportes para la Gestión Curricular	Ciencias	D-7-65	345	Kapelusz	1999	171
950-07-0428-5	Crónica de una muerte Anunciada	Novela	D-456'	9897	Sudamericana'	1992	193

Figura 1. Relación LIBROS

#### 1. Conceptos de Dominios, tuplas, atributos y relaciones

Un **dominio** D es un conjunto de valores atómicos. Por atómico queremos decir que cada valor del dominio es indivisible. Un método común de especificación de los dominios consiste en especificar un tipo de datos al cual pertenecen los valores que constituyen el dominio. También resulta útil especificar un nombre para el dominio que ayude a interpretar sus valores. Describimos a continuación algunos ejemplos:

Promedios de notas: Valores posibles de los promedios de notas calculados, cada uno debe ser un valor entre 0 y 10.

Edades de los empleados: Edades posibles de los empleados de una compañía; cada una debe ser un valor entre 16 y 80 años de edad.

Departamentos académicos: El conjunto de departamentos académicos, como Ciencias de la Computación, Economía, Física, etc. de una universidad.

Los anteriores son definiciones lógicas de dominios. También debe especificarse un tipo de datos o formato para cada dominio. Por ejemplo, se puede declarar el tipo de datos del dominio Promedios de notas son números reales inferiores a 10 y superiores a 0. El tipo de datos de Edades\_de\_empleados es un numero entero entre 16 y 80. En el caso de Departamentos\_académicos, el tipo de dato es el conjunto de cadenas de caracteres que representan nombres o códigos válidos de departamentos.

Por lo tanto, un dominio debe tener un nombre, un tipo de datos y un formato. También puede incluirse información adicional para interpretar los valores de un tipo de datos, por ejemplo, un dominio numérico de Pesos\_de\_personas deberá especificar las unidades de medición libras o kilogramos. A continuación definiremos el concepto de esquemas de relación, que describe la estructura de una relación.

Un **esquema de relación**  $R$ , denotado por  $R(A_1, A_2, A_3, \dots, A_n)$ , se compone de un nombre de relación  $R$ , y una lista de atributos  $A_1, A_2, A_3, \dots, A_n$ . Cada atributo  $A_i$  es el nombre de un papel desempeñado por algún dominio  $D$  en el esquema  $R$ . Se dice que  $D$  es el dominio de  $A_i$  y se denota con  $\text{dom}(A_i)$ . Un esquema de relación sirve para describir una relación;  $R$  es el nombre de la relación. El grado de la relación es el número de atributos,  $n$ , de su esquema de relación.

El siguiente es un esquema de relación para la relación de grado 9, que describe libros de una biblioteca:

LIBROS (ISBN,            Nombre, Genero,    Estado, Sig\_Top, Nro\_I, Editorial, Año\_de\_Ed, Cant\_pag)

Una **relación** (o instancia de relación)  $r$  del esquema de relación  $R(A_1, A_2, A_3, \dots, A_n)$ , denotado también por  $r(R)$ , es un conjunto de  $n$ -tuplas  $r = \{ t_1, t_2, t_3, \dots, t_n \}$ . Cada  $n$ -tupla  $t$  es una lista ordenada de  $n$  valores  $t = \langle v_1, v_2, \dots, v_n \rangle$ , donde cada valor  $v_i$  es un elemento de  $\text{dom}(A_i)$  o bien un valor nulo especial.

La Figura 2 muestra un ejemplo de una relación LIBROS, que corresponde al esquema LIBROS que acabamos de especificar. Cada tupla de la relación representa una entidad libro en particular. Presentamos la relación en forma de tabla, en la que cada tupla aparece como una fila y cada atributo corresponde a una cabecera de columna que indica un papel o interpretación de los valores en esa columna. Los valores nulos representan atributos cuyos valores se desconocen o no existen para algunas tuplas LIBROS individuales.

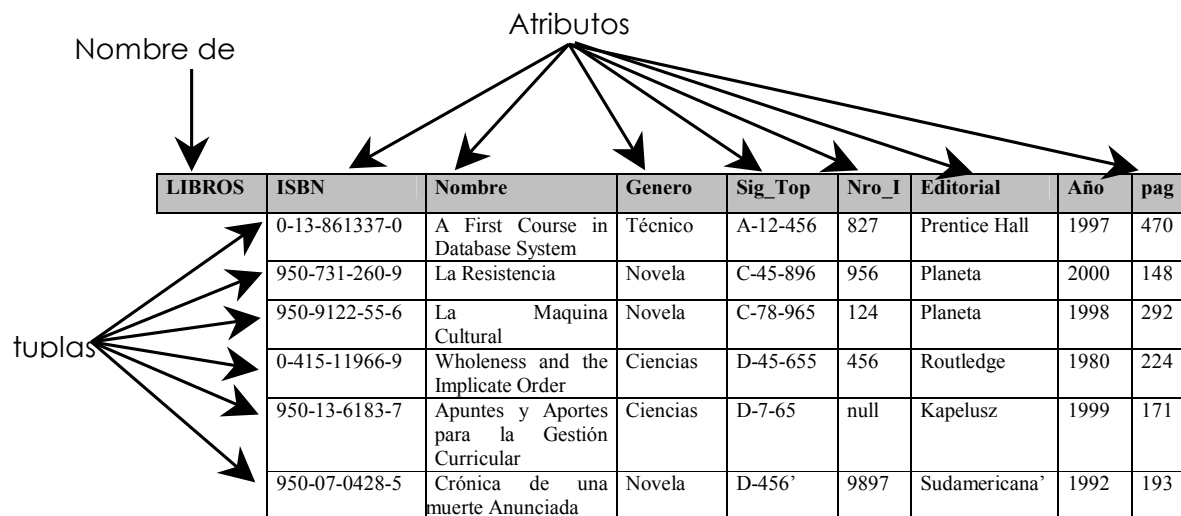


Figura 2. Atributos y tuplas de una relación LIBROS

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

## 2.Características de las relaciones

La primera definición de relación implica ciertas características que distinguen a una relación de un archivo. A continuación analizaremos algunas de estas características:

**Orden de las tuplas en una relación.** Una relación se define como un conjunto de tuplas. Matemáticamente, los elementos de un conjunto no están ordenados, por lo tanto, las tuplas de una relación no tienen un orden específico. En cambio, los registros de un archivo se almacenan físicamente en el disco, de modo que siempre existe un orden entre ellos.

El ordenamiento de las tuplas no forma parte de la definición de una relación, porque la relación intenta representar los hechos en un nivel lógico o abstracto. Podemos especificar muchos ordenamientos lógicos en una relación; por ejemplo, la tuplas de la relación LIBROS se podrían ordenar lógicamente según los valores de Nombre, o según Editorial, o según otro atributo.

Orden de los valores dentro de una tupla, y definición alternativa de relación. En un nivel lógico, el orden de los atributos y de sus valores en realidad no es importante en tanto se mantenga la correspondencia entre atributos y valores. Un esquema de relación R se puede definir también de la siguiente forma:  $R = \{ A_1, A_2, A_3, \dots, A_n \}$ , es un conjunto de atributos y una relación  $r(R)$  es un conjunto finito de transformaciones  $r = \{ t_1, t_2, t_3, \dots, t_n \}$ , donde cada tupla  $t_i$  es una transformación de R a D, y D es la unión de los dominios de los atributos; esto es,  $D = \text{dom}(A_1) \cup \text{dom}(A_2) \cup \dots \cup \text{dom}(A_n)$ . Según esta definición,  $t(A_i)$  debe estar en el  $\text{dom}(A_i)$ , para cada transformación t en r. Cada transformación  $t_i$  se denomina tupla.

De acuerdo con esta definición, podemos considerar a una tupla como un conjunto de pares ( $\langle \text{atributo} \rangle, \langle \text{valor} \rangle$ ) donde cada par da el valor de la transformación de un atributo  $A_i$  y un valor  $v_i$  de  $\text{dom}(A_i)$ . El ordenamiento de los atributos no es importante, porque el nombre del atributo aparece junto con su valor. Según esta definición, las dos tuplas de la Figura 3 son idénticas. Esto tiene sentido a nivel abstracto o lógico, ya que en realidad no existe ninguna razón para preferir que un valor de atributo aparezca antes que otro en una tupla.

Cuando una relación se implementa en forma de archivo, los atributos pueden ordenarse físicamente como campos dentro de un registro.

$t = \langle (\text{ISBN}, 0-415-11966-9), (\text{Nombre}, \text{Wholeness and the Implicate Order}), (\text{Genero}, \text{Ciencias}), (\text{Sig\_Top}, \text{D-45-655}), (\text{Nro\_I}, 456), (\text{Editorial}, \text{Routledge}), (\text{Año}, 1980), (\text{Paginas}, 224) \rangle$
$t = \langle (\text{Sig\_Top}, \text{D-45-655}), (\text{Nro\_I}, 456), (\text{Nombre}, \text{Wholeness and the Implicate Order}), (\text{Genero}, \text{Ciencias}), (\text{Editorial}, \text{Routledge}), (\text{Año}, 1980), (\text{Paginas}, 224), (\text{ISBN}, 0-415-11966-9) \rangle$

Figura 3. Dos tuplas idénticas cuando el orden de los atributos y de los valores no forma parte de la definición de una relación.

**Valores de Tuplas:** Cada valor en una tupla es un valor atómico; esto es, no es divisible en componentes en lo que respecta al modelo relacional. Por ello no se permiten atributos compuestos. Gran parte de la teoría que apoya al modelo relacional se desarrolló tomando en cuenta esta suposición, conocida como suposición de primera forma normal.

Puede ser que los valores de algunos atributos dentro de una tupla en particular sean desconocidos o no se apliquen a esa tupla. En estos casos se utiliza un valor especial, llamado nulo. Por ejemplo, en la Figura 2 una tupla de libro tiene un valor nulo como  $\text{Num\_I}$  (número de inventario), debido a que el libro no ha sido inventariado aún. En otro caso el nulo podría indicar que no se cuenta con dicha información, por ejemplo si un libro no ha sido editado por ninguna editorial conocida.

En general, podemos tener varios tipos de valores nulos, como “valor desconocido”, “atributo no aplicable a esa tupla”, o esta “tupla no tiene valor para este atributo”. De hecho, algunas implementaciones establecen diferentes códigos para los distintos tipos de valores nulos.

Cabe señalar que algunas relaciones pueden representar hechos acerca de entidades, en tanto que otras pueden representar hechos sobre vínculos. Por ejemplo, un esquema de relación ( $\text{ISBN}$ ,  $\text{NomAutor}$ ) establece el autor de cada libro; una tupla de esta relación relaciona un libro con uno de los autores del libro. De este modo, el modelo relacional representa hechos acerca de entidades y de inter-relaciones de un MER uniformemente como relaciones (en el modelo relacional).

### **3. Restricción de Integridad de Claves**

Una relación se define como un conjunto de tuplas. Por definición, todos los elementos de un conjunto son distintos; por lo 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 de la relación  $r$  de  $R$  con la misma combinación de valores para esos atributos. Supongamos que denotamos un subconjunto así de atributos con  $SC$ ; entonces, para cualesquiera dos tuplas distintas  $t_1$  y  $t_2$  en un ejemplar de relación  $r$  de  $R$ , tenemos la siguiente restricción  $t_1[SC] \neq t_2[SC]$ .

Todo conjunto de atributos  $SC$  de este tipo es una superclave del esquema de relación  $R$ . Toda relación tiene por lo menos una **superclave**: el conjunto de todos los atributos. Sin embargo, una superclave puede tener atributos redundantes, así 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 superclave de  $R$ . Por tanto, una clave es una superclave mínima; una superclave a la cual no podemos quitarle ningún atributo sin que deje de cumplirse **la restricción de unicidad**.

Como ejemplo consideremos la relación LIBROS de la Figura 1. El conjunto de atributos ISBN es una clave de LIBROS porque no puede haber dos tuplas de estudiantes que tengan el mismo valor de ISBN. Cualquier conjunto de atributos que contenga a ISBN –por ejemplo {ISBN, Nombre, Editorial}– es una superclave.

El valor de un atributo clave puede servir para identificar de manera única una tupla de la relación. Por ejemplo, el valor de ISBN 0-415-11966-9 sirve para identificar de manera única a la tupla correspondiente al libro “Wholeness and the Implicate Order”. Observe que 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 todas las relaciones posibles del esquema. La clave se determina a partir del significado de los atributos en el esquema de la relación; por ende, la propiedad no varía con el tiempo; debe seguir siendo válida aunque insertemos tuplas nuevas a la relación.

En general, un esquema de relación puede tener más de una clave. En tal caso, cada una de ellas se denomina clave candidata. Por ejemplo, la relación AUTOMOVIL tiene dos claves candidatas: Numero de Matricula, y por otro lado Numero de Serie de Motor. Es común designar a una de las claves candidatas como la clave primaria de la relación. Esta es la clave candidata cuyos valores sirven para identificar las tuplas en la relación. Adoptaremos la convención de subrayar los atributos que forman la clave primaria de un esquema de relación. Cabe señalar que, cuando un esquema tiene varias claves candidatas, la elección de una clave primaria es arbitraria; sin embargo, casi siempre es mejor escoger una clave primaria con un solo atributo o un número reducido de atributos.

#### **4. Esquemas de Bases de Datos.**

Hasta ahora hemos visto relaciones y esquemas de relaciones individuales. Pero, de hecho, una base de datos relacional suele contener muchas relaciones y en éstas las tuplas están relacionadas de diversas maneras. En esta sección definiremos una base de datos relacional y un esquema de base de datos relacional. Un esquema de base de datos relacional es un conjunto de esquemas de relaciones  $S = \{R_1, R_2, \dots, R_n\}$  y un conjunto de restricciones. Un ejemplar de base de datos relacional BD de S es un conjunto de ejemplares de relaciones  $BD = \{r_1, r_2, \dots, r_n\}$  tal que cada ejemplar de  $r_i$  es un ejemplar de  $R_i$ .

La Figura 4 muestra el esquema de base de datos relacional COMPAÑÍA. Un ejemplar de esta base de datos contendrá una colección de tuplas para cada relación de cada esquema.

##### DEPARTAMENTOS

<u>Nombre</u>	Número	NumDeEmpleados	DNIGerente
---------------	--------	----------------	------------

##### PROYECTOS

<u>Nombre</u>	<u>Número</u>	Lugar
---------------	---------------	-------

##### EMPLEADOS

<u>DNI</u>	Nombre	Apellido	Sexo	FechaN	NombreDpto	DNISupervisor
------------	--------	----------	------	--------	------------	---------------

##### TRABAJA\_EN

<u>DNIEmpleado</u>	<u>NombreDepto</u>
--------------------	--------------------

##### CONTROLA

<u>NombreProyecto</u>	<u>NumeroProyecto</u>	<u>NombreDepartamento</u>
-----------------------	-----------------------	---------------------------

Figura 4. Restricciones de Integridad Referencial en el esquema de base de datos COMPAÑÍA

En la figura 4 el atributo NOMBRE de DEPARTAMENTOS y NOMBREDEPARTAMENTO de CONTROLA representan el mismo concepto del mundo real: el nombre otorgado a un departamento. Permitiremos que un atributo que represente el mismo concepto del mundo real tenga nombres que no pueden ser idénticos en diferentes relaciones. De manera similar, permitiremos que diferentes atributos que representen diferentes conceptos tengan el mismo nombre en relaciones distintas. Por ejemplo, hemos utilizado el atributo NOMBRE tanto en EMPLEADOS como en PROYECTOS pero representan conceptos distintos del mundo real: nombres de empleados, y nombres de proyectos.

#### **5. Restricción de Integridad de Entidades**

La restricción de *integridad de entidades* establece que ningún valor de clave primaria puede ser nulo. Esto es porque el valor de la clave primaria sirve para identificar las tuplas individuales en 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.

## 6. Restricción de Integridad Referencial

Las restricciones de clave y de integridad de entidades se especifican sobre relaciones individuales. La *restricción de integridad referencial* se especifica entre dos relaciones y sirve para mantener la consistencia entre tuplas de 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 "otra" relación. Por ejemplo, si cada empleado trabaja en un departamento, el valor de cada tupla de empleado deberá coincidir con el valor de un Nombre de departamento de la relación departamento. Esta restricción referencial está marcada con línea punteada en la Figura 5.

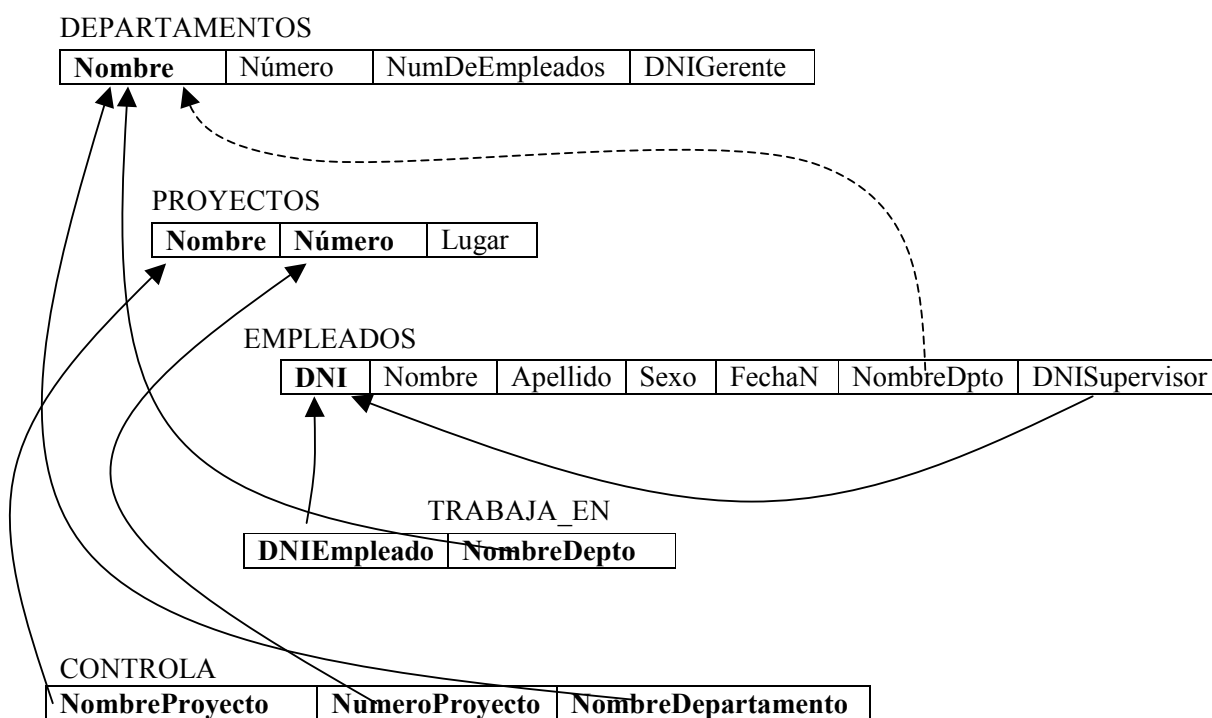


Figura 5. Restricciones de Integridad Referencial en el esquema de base de datos COMPAÑÍA

En una base de datos con muchas relaciones, suele haber muchas restricciones de integridad referencial. Para especificar restricciones es preciso, primero, comprender con claridad el significado o papel que cada uno de los conjuntos de atributos desempeña en los diversos esquemas de relaciones de la base de datos.

Las restricciones de integridad referencial casi siempre surgen de los vínculos entre las entidades representadas en los esquemas de relación. Por ejemplo, en la relación **EMPLEADOS** el atributo **NombreDpto** se refiere al departamento para el cual trabaja un

empleado; por lo tanto, designamos a NombreDpto como clave externa de EMPLEADOS, con referencia a la relación DEPARTAMENTOS. Esto significa que un valor para NombreDpto en cualquier tupla t de la relación EMPLEADOS deberá coincidir con un valor de la clave primaria de DEPARTAMENTOS (el atributo NOMBRE en DEPARTAMENTOS), o el valor de NombreDpto podría ser nulo si el empleado no pertenece a ningún departamento.

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

En la Figura 5 se representó diagramáticamente las restricciones de integridad referencial trazando un arco dirigido de cada clave externa a la relación a la cual hace referencia. Para mayor claridad la punta de la flecha puede apuntar a la clave primaria de la relación referida.

En un sistema relacional, el lenguaje de definición de datos (DDL) cuenta con mecanismos para especificar los diversos tipos de restricciones para que el SGBD pueda imponerlas automáticamente.

### **7. Restricciones de integridad de datos: Reglas de Negocio**

Los tipos de restricciones que hemos visto no incluyen una amplia clase de restricciones generales, a veces llamada restricciones de integridad semántica o restricciones específicas. Ejemplos de tales restricciones son, suponiendo que el esquema EMPLEADOS tuviera un atributo Salario: “el salario de un empleado no debe exceder el salario de su supervisor”. Otro ejemplo, “el número máximo de proyectos que controla un departamento es 20”. Otro, que “el número máximo de empleados de un departamentos es 50”.

### **8. Del Modelo Entidad Relación a Esquemas de Base de Datos Relacionales.**

Habíamos mencionado en la unidad 2 que es posible construir un esquema conceptual traduciéndolo desde el modelo de datos de alto nivel (como es el MER) a un modelo de datos más cercano a la implementación (como lo es un modelo Relacional). Este paso se de **transformación de modelos de datos** tiene como resultado un esquema de base de datos especificado en el modelo de datos de implementación del SGBD. En particular, el MER que representemos se traducirá al modelo de datos relacional.

#### **La Derivación.**



Habíamos mencionado previamente que el modelo relacional representa una base de datos como una colección de tablas. Dado entonces, un conjunto de entidades y relaciones pertenecientes a una representación del MER obtendremos un conjunto de las tablas correspondientes a la información contenida en el MER.

### 8.1. Derivación de Entidades.

Comenzaremos la derivando una tabla por cada entidad incluida en el MER. Siguiendo con el ejemplo de la base de datos COMPAÑÍA de la unidad 2 –ver MER en Figura 6-, obtenemos las siguientes tablas, a partir de las tres entidades principales:

DEPARTAMENTOS (Nombre, Número, NumDeEmpleados)

PROYECTOS (Nombre, Número, Lugar)

EMPLEADOS (DNI, Nombre, Apellido, Sexo, FechaN)

**Notación:** la tabla que representa la relación en el modelo relacional estará representada por su nombre en mayúsculas y entre paréntesis escribiremos los nombres de los atributos cuya inicial escribiremos en mayúsculas.

### 8.2. Derivación de Relaciones.

En el caso de la derivación de relaciones el proceso para la obtención de tablas no es tan simple, y debemos analizar fundamentalmente cuales son las cardinalidades que han modelado

#### 8.2.1. Derivación de Relaciones Binarias con cardinalidad 1:1.

En nuestra base de datos COMPAÑÍA contamos con un ejemplo de una relación 1:1 es la relación DIRIGE. Es claro que todo departamento tiene un gerente, y que no todo empleado es gerente, por ello en estos casos optaremos por incluir la clave de la entidad EMPLEADO en la derivación de la entidad DEPARTAMENTO. En otras palabras, la información del gerente de un departamento estará almacenada como un atributo de la entidad DEPARTAMENTO.

Agregamos esta información de la siguiente manera:

DEPARTAMENTOS (Nombre, Número, NumDeEmpleados, DNIGerente)

PROYECTOS (Nombre, Número, Lugar)

EMPLEADOS (DNI, Nombre, Apellido, Sexo, FechaN)

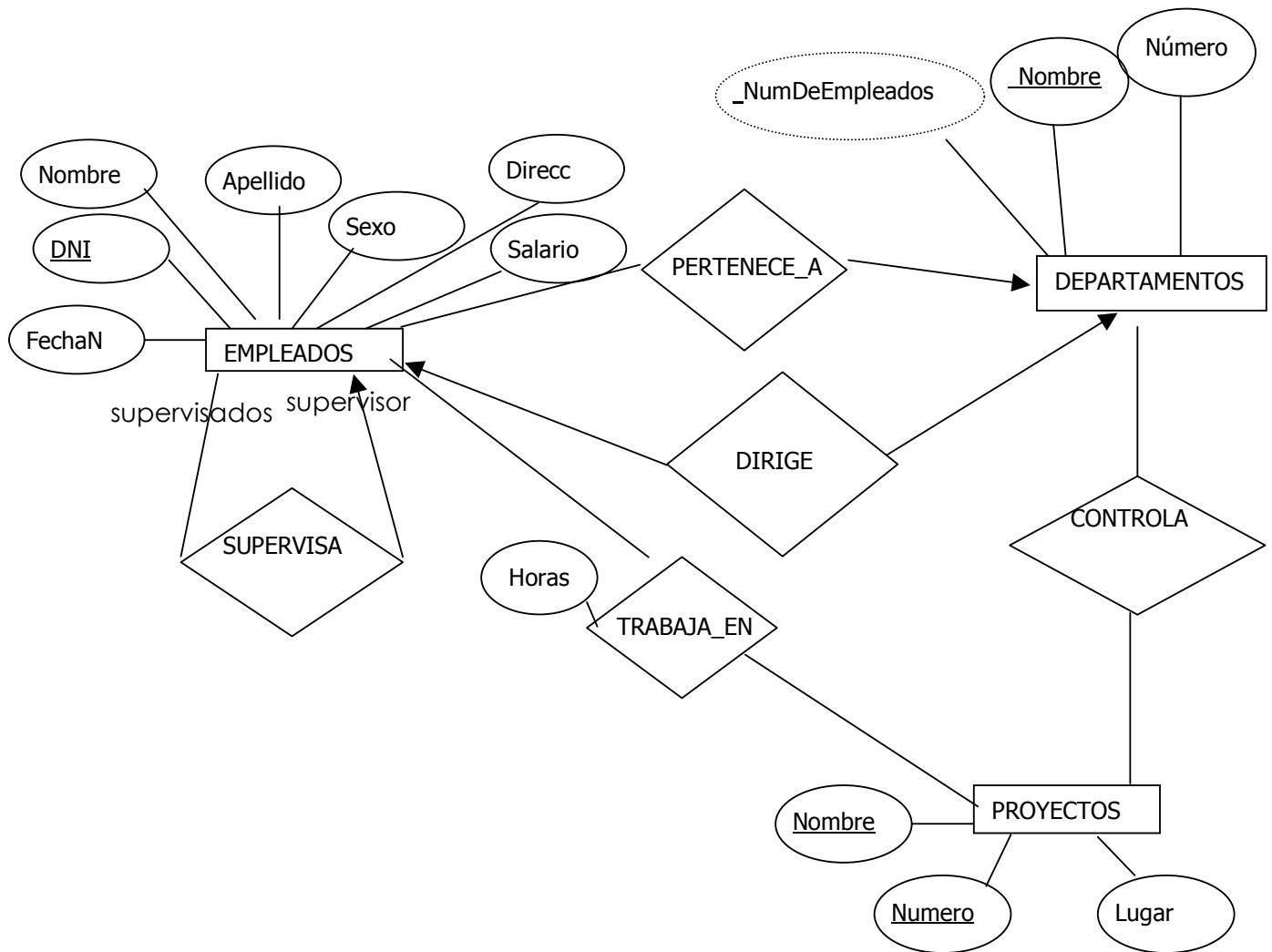


Figura 6. Modelo Entidad Relación de la base de datos COMPAÑÍA

### 8.2.2. Derivación de Relaciones Binarias con cardinalidad 1:N.

La derivación consiste en incluir un nuevo atributo en la derivación de la entidad que representa el extremo N de la relación, dicho atributo (o atributos) consistirá en la clave de la entidad que representa el extremo 1.

PERTENECE\_A es una relación con cardinalidad 1:N, el extremo con cardinalidad N refiere a la entidad EMPLEADOS. Esta relación representa el hecho que todo empleado pertenece a un departamento y que todo departamento tiene empleados. Procediendo como detallamos arriba, incluiremos en la entidad EMPLEADOS, un atributo *NombreDepto* que contendrá información

sobre el departamento en el cual trabaja un empleado. Note que en nuestro ejemplo la clave de la entidad DEPARTAMENTOS consiste en su Nombre. Entonces:

DEPARTAMENTOS (Nombre, Número, NumDeEmpleados, DNIGerente)

PROYECTOS (Nombre, Número, Lugar)

EMPLEADOS (DNI, Nombre, Apellido, Sexo, FechaN, NombreDpto)

### 8.2.3. Derivación de Relaciones Binarias con cardinalidad M:N.

Las relaciones binarias con cardinalidad M:N, se derivan en nuevas tablas. Así la relación TRABAJA\_EN entre EMPLEADOS y PROYECTOS, se deriva en una nueva tabla:

DEPARTAMENTOS (Nombre, Número, NumDeEmpleados, DNIGerente)

PROYECTOS (Nombre, Número, Lugar)

EMPLEADOS (DNI, Nombre, Apellido, Sexo, FechaN, NombreDpto)

TRABAJA\_EN (DNI\_empleado, Nombre\_Proyecto)

Resta analizar el caso de la relación CONTROLA, relación binaria de cardinalidad 1:N entre DEPARTAMENTO y PROYECTO, estamos en el caso 4.2.3. por ello incorporamos la clave del extremo 1 de la relación en la derivación del extremo N.

DEPARTAMENTOS (Nombre, Número, NumDeEmpleados, DNIGerente)

PROYECTOS (Nombre, Número, Lugar)

EMPLEADOS (DNI, Nombre, Apellido, Sexo, FechaN, NombreDpto)

TRABAJA\_EN (DNI\_empleado, Nombre\_Proyecto)

CONTROLA (NombreProyecto, NumeroProyecto, NombreDepartamento)

Para finalizar con nuestro ejemplo, necesitamos analizar como derivar la relación SUPERVISA, que asocia la entidad EMPLEADOS consigo misma, es una relación uno muchos, por lo tanto la clave del uno la derivamos en el mucho (esto es la misma entidad).

La derivación final obtenida es:

DEPARTAMENTOS (Nombre, Número, NumDeEmpleados, DNIGerente)

PROYECTOS (Nombre, Número, Lugar)

EMPLEADOS (DNI, Nombre, Apellido, Sexo, FechaN, NombreDpto, DNISupervisor)

TRABAJA\_EN (DNI\_empleado, Nombre\_Proyecto)

CONTROLA (NombreProyecto, NumeroProyecto, NombreDepartamento)

#### **8.2.4. Derivación de Relaciones Ternarias.**

En el caso de la derivación de relaciones ternarias, se deriva la relación en una nueva tabla, en la misma forma que en caso 8.2.3. Por supuesto, las claves de las tres entidades involucradas serán parte del esquema de la tabla derivada.