Unidad IV. Diseño de un Sistema de BD
M. en C. Euler Hernández Contreras

4.1 Modelo Relacional
4.1.1 Introducción
4.1.2 Componentes del modelo relacional (Estructura de datos, Manipulación de datos e integridad de datos).
4.1.3 Llaves relacionales (Llave primaria, llave compuesta, llave foránea, llave candidata).
4.1.4 Propiedades de una relación.
4.1.5 Reglas de integridad (nulo, integridad de entidades, integridad referencial).
4.2 Transformación del Diagrama ER al Relacional.
4.3 Normalización
4.3.1 Anomalías y Relaciones bien estructuradas.
4.3.2 Pasos en la normalización
4.3.3 Dependencias Funcionales y Llaves
4.3.4 Formas Normales (1FN, 2FN, 3FN, BCFN).
4.3.5 Otras Formas Normales
4.3.6 Ejemplos.

Referencia Bibliográfica

1. Michael V. Mannino. Administración de bases de datos, diseño y desarrollo de aplicaciones, Tercera Edición. Mc Graw Hill Interamericana, México 2007, 712 págs.

2. Date C. J. Introducción a los Sistemas de Bases de Datos, Séptima Edición. Pearson Educación de México, México 2001.

3. Hoffer A. Jeffrey, Prescott Mary B., Topi Heikki. Modern Database Management, Ninth Edition, Pearson/Prentice, Estados Unidos 2009.

4. Elmasri Ramez, Navathe Shamkant B. Fundamentos de Sistemas de Bases de Datos, Quinta Edición. Pearson/Addison Wesley, Madrid España 2007, págs. 988 ISBN: 978-84-7829-085-7

5. Ramakrishnan Raghu, Gehrke Johannes. Sistemas de Gestión de Bases de Datos,Tercera Edición. McGraw-Hill/Interamericana de España, Madrid España 2007, págs. 654 ISBN: 978-84-481-5638-1

6. Ricardo Catherine M. Bases de Datos. Mc Graw Hill, México D.F. 2009, págs. 642. ISBN: 978-970-10-7275-2

4.1 Introducción

El modelo Relacional fue introducido en 1970 por E.F. Codd.

4.2 Conceptos Básicos

En el modelo relacional representamos los datos en forma de tablas y éste consiste en tres componentes:

a) Estructura de datos: Los datos son organizados en forma de tablas con filas y columnas.
b) Manipulación de datos: Podemos realizar operaciones sobre los datos utilizando el lenguaje SQL (*Structure Query Language*) sobre los datos almacenados en las relaciones.
c) Integridad de datos: Nos permite incluir las reglas del negocio de una organización, cuando estos son manipulados.

4.2.1 Definiciones básicas

Algunas definiciones se comentan a continuación (Ver Figura 26):

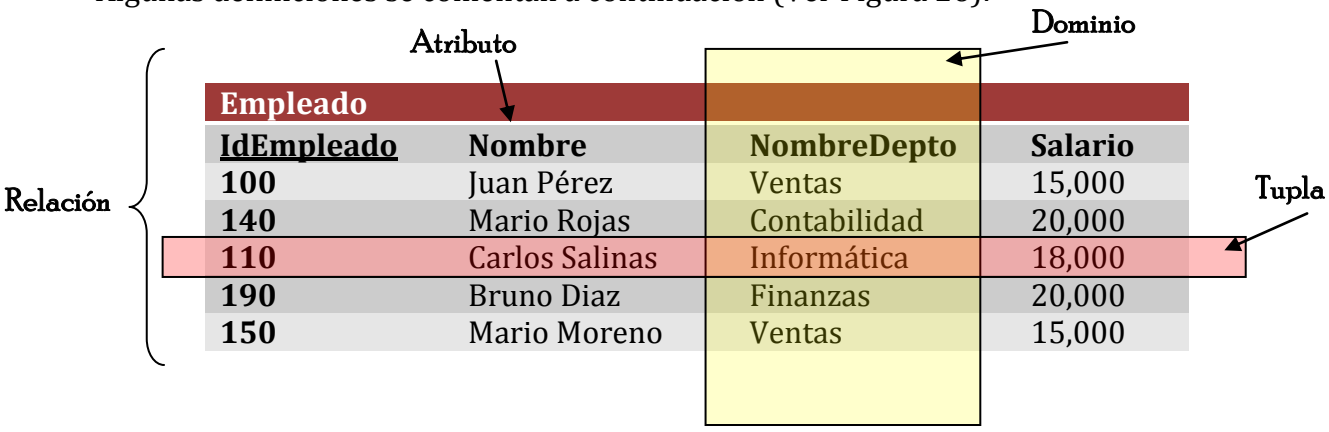


Figura 26. Algunas definiciones en el Modelo Relacional

Una *relación* es una tabla con columnas y filas.
Un *atributo* es el nombre de una columna de una relación.
Un *dominio* es el conjunto de valores legales de uno o varios atributos.
Una *tupla* es una fila de una relación.
El *grado* de una relación es el número de atributos que contiene.
La *cardinalidad* de una relación es el número de tuplas que contiene.
Esquema conceptual es la representación de una estructura lógica de la base de datos.

4.2.2 Llaves Relacionales

Una *llave primaria (PK- primary key)* es un atributo o combinación de atributos que identifican de manera única cada registro en una relación. Por ejemplo, la llave primaria para la relación empleado (Ver Figura 27)

Asociado(idAsociado, nombre, apellido_Paterno, apellidoMaterno, género, salario)

Figura 27. Llave primaria

Una *llave compuesta* es una llave primaria que consiste de más de un atributo (Ver Figura 28).

Socio(idSocio, email, nombre, dirección, tel)

Figura 28. Llave primaria Compuesta

Una *llave foránea (ajena/externa –FK foreign key-)* es un atributo (posiblemente compuesto) dentro de una relación de la BD que sirve como llave primaria de otra relación en la misma BD. Por ejemplo, consideraremos las relaciones Empleado y Departamento (Ver Figura 29).

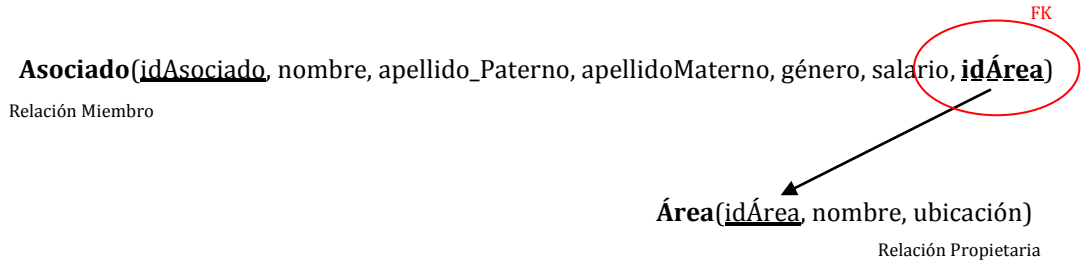


Figura 29. Llave Foránea

4.3 Propiedades de las relaciones

Las relaciones tienen las siguientes características:

- Cada relación (tabla) en una BD, tiene un nombre y éste es distinto del nombre de todas las demás.
- Los valores de los atributos son atómicos (simples) en cada tupla, es decir, cada atributo toma un solo valor. Se dice que las relaciones están normalizadas.
- No hay dos atributos que se llamen igual.
- El orden de los atributos no importa, es decir los atributos no están ordenados.
- Cada tupla (fila) es distinta de las demás, es decir, no hay tuplas duplicadas.
- El orden de las tuplas no importa, es decir, las tuplas no están ordenadas.

4.4 Reglas de Integridad

Una vez definida la estructura de datos del modelo relacional, pasamos a estudiar las reglas de integridad que los datos almacenados en dicha estructura deben cumplir para garantizar que son correctos.

Al definir cada atributo sobre un dominio se impone una restricción sobre el conjunto de valores permitidos para cada atributo. A este tipo de restricciones se les denomina restricciones de dominios.

Hay además dos reglas de integridad muy importantes que son restricciones que se deben cumplir en todas las bases de datos relacionales y en todos sus estados o instancias (las reglas se deben cumplir todo el tiempo). Estas reglas son:

a) Regla de integridad de entidades.
b) Regla de integridad referencial.

Antes de definirlas, es preciso conocer el concepto de nulo.

4.4.1 Nulo

Cuando en una tupla un atributo es desconocido, se dice que es *nulo*.

Un *nulo* no representa el valor cero ni la cadena vacía, éstos son valores que tienen significado. El nulo implica ausencia de información, bien porque al insertar la tupla se desconocía el valor del atributo, o bien porque para dicha tupla el atributo no tiene sentido.

4.4.2 Regla de integridad de entidades

La primera regla de integridad se aplica a las claves primarias de las relaciones base:
"Ninguno de los atributos que componen la clave primaria puede ser nulo".

Por definición, una clave primaria es un identificador irreducible que se utiliza para identificar de modo único las tuplas. ¿Qué es *irreducible*? significa que ningún subconjunto de la clave primaria sirve para identificar las tuplas de modo único. Si se permite que parte de la clave primaria sea nula, se está diciendo que no todos sus atributos son necesarios para distinguir las tuplas, con lo que se contradice la irreducibilidad.

Nótese que esta regla sólo se aplica a las relaciones propietarias y a las claves primarias, no a las claves alternativas.

4.4.3 Regla de integridad referencial

La segunda regla de integridad se aplica a las claves ajenas: *"si en una relación hay alguna clave ajena (foránea), sus valores deben coincidir con valores de la clave primaria a la que hace referencia, o bien, deben ser completamente nulos."*

La regla de integridad referencial se enmarca en términos de estados de la base de datos: indica lo que es un estado ilegal, pero no dice cómo puede evitarse. La cuestión es ¿qué hacer si estando en un estado legal, llega una petición para

realizar una operación que conduce a un estado ilegal? Existen dos opciones: rechazar la operación, o bien aceptar la operación y realizar operaciones adicionales compensatorias que conduzcan a un estado legal.

4.4.4 Reglas del negocio

Además de las dos reglas de integridad anteriores, los usuarios o los administradores de la base de datos pueden imponer ciertas restricciones específicas sobre los datos, denominadas reglas de negocio.

Por ejemplo, si en una oficina de la empresa inmobiliaria sólo puede haber hasta veinte empleados, el SGBD debe dar la posibilidad al usuario de definir una regla al respecto y debe hacerla respetar. En este caso, no debería permitir dar de alta un empleado en una oficina que ya tiene los veinte permitidos.

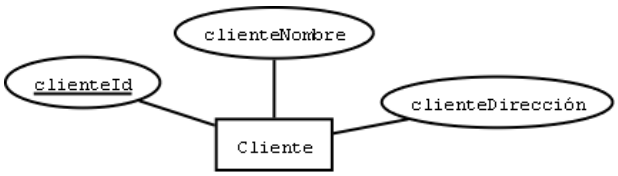
Reglas de negocio (*business rules*): Son descripciones narrativas de políticas, procedimientos o principios dentro de una organización.

4.5Mapeo del Diagrama Entidad Relación al Modelo Relacional

Paso 1: Mapeo de Entidades
Cada entidad en un Diagrama Entidad Relación (DER) es transformado en una Relación. El nombre que se le asigna a la relación es generalmente el mismo nombre que tiene la entidad.

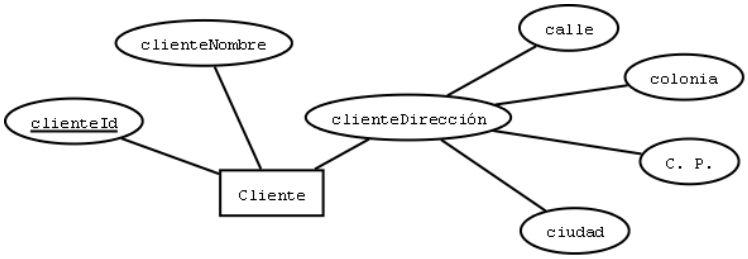
Cada atributo en una entidad se convierte en atributo de la relación.

El **atributo identificador** de la entidad se convierte en la llave primaria de la relación correspondiente.



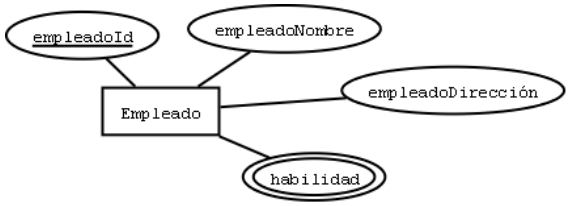
Cliente (ClienteId, clienteNombre, clienteDirección)

Atributos Compuestos: Cuando una entidad tiene atributos compuestos, solo los componentes del atributo compuesto son incluidos en la nueva relación.



Cliente(ClienteId, clienteNombre,**calle,colonia,c.p., ciudad**)

Atributos Multivalor: Cuando una entidad contiene atributos multivalor, dos nuevas relaciones son creadas. La primera relación contiene todos los atributos de la entidad excepto el atributo multivalor. La segunda relación contiene dos atributos que forman la llave primaria (llave primaria compuesta) de la segunda relación. El primero de estos atributos es la llave primaria de la primera relación, la cual se convierte en la llave foránea en la segunda relación. El segundo atributo es el atributo multivalor. El nombre de la segunda relación debe ser representativo del atributo multivalor.

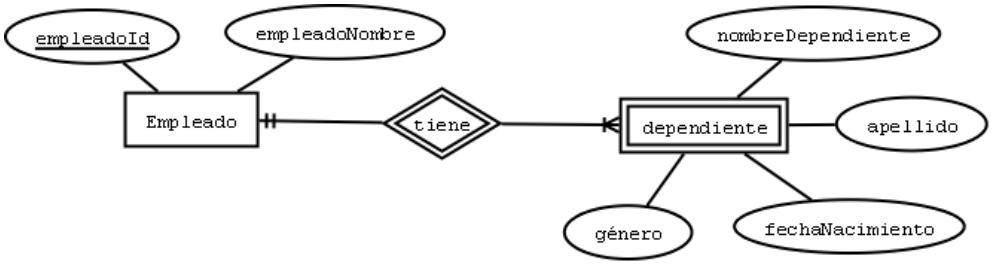


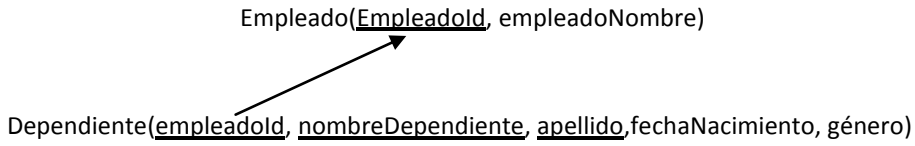
Empleado(EmpleadoId, empleadoNombre, empleadoDirección)

Empleado_Habilidad(empleadoId,Habilidad)

Paso 2: Mapeo de Entidades Débiles

Para cada entidad débil, se crea una nueva relación e incluye todos los atributos simples (o atributos simples de atributos compuestos) como atributos de esta relación. Posteriormente incluye la llave primaria de la entidad fuerte o relación identificada como llave foránea en esta nueva relación. La llave primaria de la nueva relación es la combinación de la llave primaria de la entidad fuerte y el identificador parcial de la entidad débil.

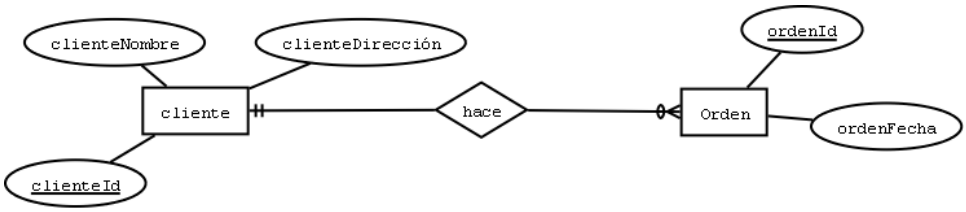


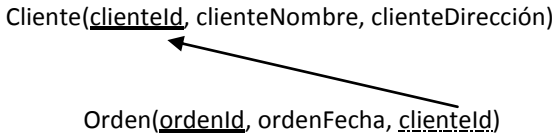


Paso 3: Mapeo de Relaciones Binarias

Mapeo de Relaciones Binarias Uno a Muchos (1:M)

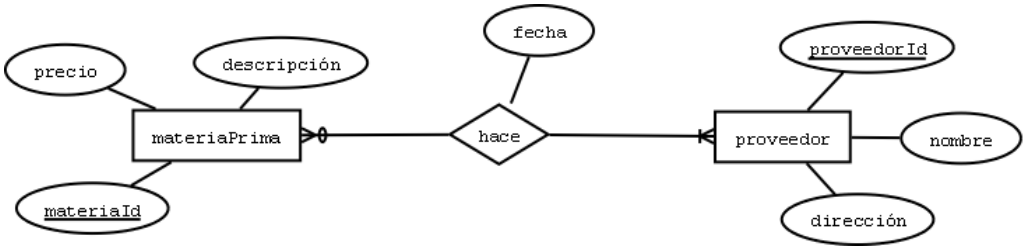
Para cada relación 1:M, primero se crea una relación para cada una de las dos entidades participantes en la relación, usando el procedimiento descrito en el paso 1. Se debe incluir la llave primaria de la entidad del lado de la cardinalidad uno (1) como llave foránea en la relación que está del lado de cardinalidad muchos (M) de la relación. En otras palabras, la llave primaria se pasa como llave foránea al lado de los muchos (M).





Mapeo de Relaciones Binarias Muchos a Muchos (M:N)

Suponga la relación entre dos entidades A y B. Para hacer el mapeo se deben generar 2 relaciones una para cada una de las entidades (A y B) y una adicional para la relación M:N entre las entidades, que llamaremos C. Se debe incluir como llaves foráneas de la relación C, la llave primaria de cada una de las dos entidades (A y B). Estos mismos atributos (llaves foráneas) se convierten en la llave primaria de la relación C, cualquier atributo asociado a la relación M:N se debe incluir en la relación C.



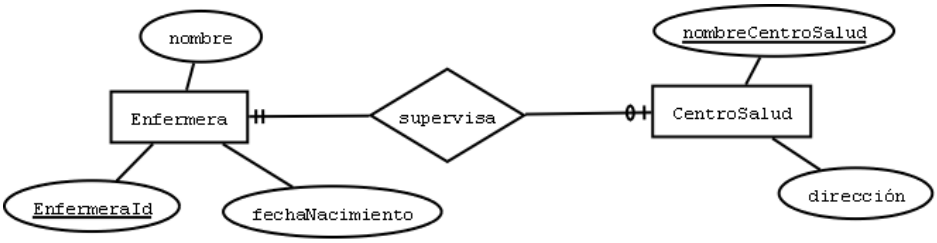
MateriaPrima(materialId, precio, descripción)

Proveedor_MateriaPrima(materialId,proveedorId, fecha)

Proveedor(proveedorId, nombre, dirección)

Mapeo de Relaciones Binarias Uno a Uno (1:1)

Para una relación binaria uno a uno (1:1) se debe crear 2 relaciones, una para cada entidad involucrada. La llave primara de una de las relaciones resultantes es incluida como llave foránea de la otra entidad. Si la relación binaria es opcional en una dirección y obligatoria hacia la otra dirección (como se muestra en el ejemplo), la llave foránea debe de ir del lado de la relación opcional, esto para evitar valores nulos.



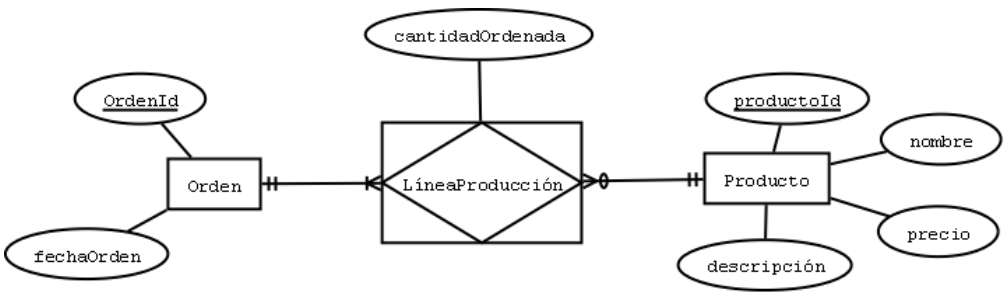
Enfermera(EnfermeraId, nombre, fechaNacimiento)

CentroSalud(nombreCentroSalud, dirección, enfermeraSupervisoraId)

Paso 4: Mapeo de Entidades Asociativas

Se crean 3 relaciones, una para cada una de las entidades, incluyendo la entidad asociativa (la cual se denomina relación asociativa al hacer el mapeo). Existen dos casos al hacer el mapeo al modelo relacional, el primero es si no existe un identificador asignado a la entidad asociativa y el segundo es si existe un identificador asignado a la entidad asociativa.

Si no existe un identificador asignado, la llave primaria de la relación asociativa se va a componer de dos atributos (llave compuesta), los cuales son las llaves primarias de las otras dos relaciones. Los atributos simples asociados a la entidad asociativa se agregan a la relación asociativa como atributos adicionales.

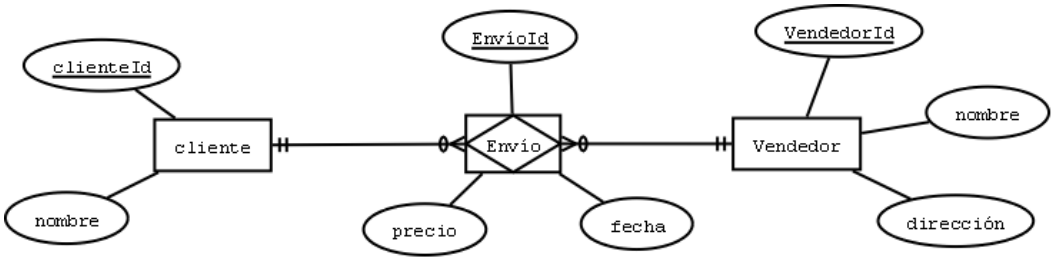


Orden(OrdenId, fechaOrden)

Línea_Producción(OrdenId,ProdudctoId, cantidadOrdenada)

Producto(ProductoId, nombre, precio, descripción)

Si existe un identificador asignado a la entidad asociativa, se crean tres relaciones, incluyendo la relación asociativa, la llave primaria de la relación asociativa va a ser el atributo identificador asociado a la misma y no será una llave compuesta como en el caso anterior. Las llaves primarias de las otras dos entidades participantes son incluidas como llaves foráneas en la relación asociativa resultante.



Cliente(clienteID, nombre)

Envío(EnvíoId,clienteID,VendedorId, precio, fecha)

Vendedor(vendedorId, nombre, dirección)

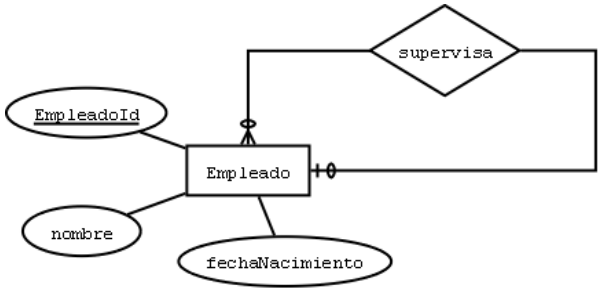
Paso 5: Mapeo de Relaciones Unarias

Las relaciones unarias, son relaciones entre instancias de una sola entidad, las relaciones unarias son también llamadas relaciones recursivas. Existen dos casos principales para estas relaciones unarias que son para cardinalidades: uno a muchos (1:M) y muchos a muchos (M:N).

Relaciones Unarias Uno a Muchos (1:M)

El mapeo de las relaciones unarias sigue el mismo procedimiento que el paso 1, es decir se crea una relación para la entidad involucrada. La llave foránea es agregada como un atributo dentro de la misma relación que hace referencia a los valores de la llave primaria (esta llave foránea debe tener el mismo dominio que la llave primaria).

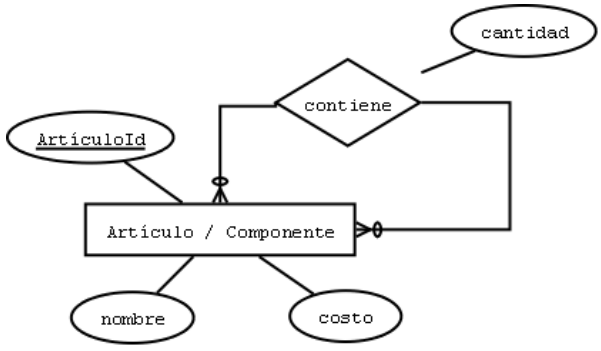
Llave foránea recursiva: es una llave foránea en una relación que hace referencia a los valores de la llave primaria de la misma relación.



Empleado(empleadoId, nombre, fechaNacimiento, SupervisorId)

Relaciones Unarias Muchos a Muchos (M:N)

Con este tipo de relación se crean dos relaciones, una para representar al tipo de entidad en la relación y la otra será una relación asociativa para representar la relación muchos a muchos (M:N). La llave primaria de la relación asociativa consistirá de dos atributos. Ambos atributos (los cuales no deben llevar el mismo nombre) toman los valores de la llave primaria de la otra relación, cualquier atributo adicional que no sea llave se incluye en la relación asociativa.

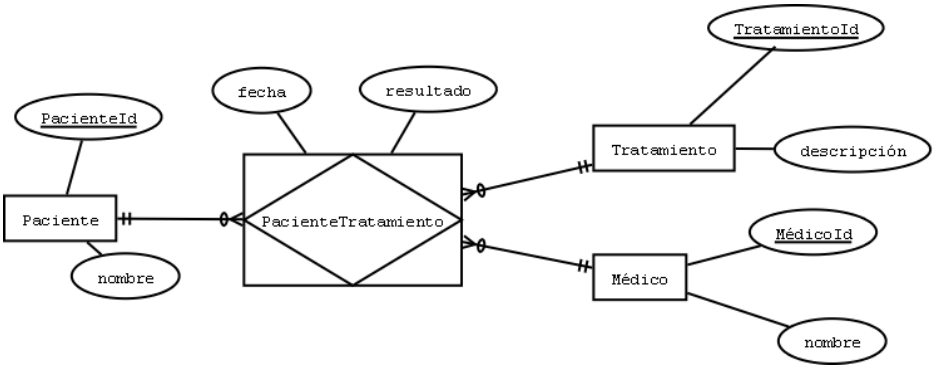


Artículo(artículoId, nombre, costo)

Componente(artículoId, componenteId, cantidad)

Paso 6: Mapeo de Relaciones Ternarias y N-arias
Una relación ternaria es una relación entre tres tipos de entidades, se sugiere que relaciones ternarias se conviertan en entidades asociativas para representar la participación de las tres entidades más apropiadamente.

Para hacer el mapeo de una entidad asociativa que liga tres entidades se deben crear tres relaciones y una relación adicional, la relación asociativa. La llave primaria de esta relación asociativa consiste de tres atributos que son las llaves primarias de las entidades participantes (en ocasiones se requieren de atributos adicionales para formar una llave primaria única). Estos atributos también son las llaves foráneas que hacen referencia a las llaves primarias de las entidades que participan en la relación. Cualquier atributo adicional asociado a la entidad asociativa es agregado a los atributos de la relación asociativa



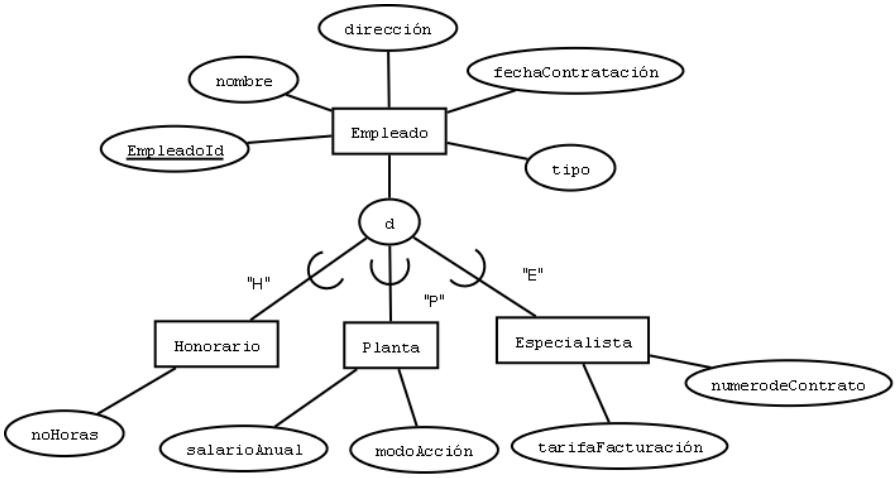
Paciente(<u>pacienteId</u>, nombre)

Médico(<u>médicoId</u>, nombre)

PacienteTratamiento(<u>pacienteId</u>, <u>fecha</u>, resultado, <u>tratamientoId</u>, <u>médicoId</u>)

Tratamiento(<u>TratamientoId</u>, descripción)

Paso 7: Mapeo de Relaciones Supertipo/Subtipo
Este tipo de transformaciones es usado para denotar el diagrama relación extendido.



Especialista(E_empleadoId, tarifaFacturación, númeroDeContrato)

Empleado(empleadoId, nombre, dirección, fechaContratación, tipo)

Honorario(H_empleadoId, noHoras)

Planta(P_empleadoId, salarioAnual, modoAcción)

4.6 Normalización

Relaciones bien estructuradas
Antes de hablar de normalización, necesitamos preguntarnos, ¿cuándo una relación se encuentra bien estructurada?

Una relación bien estructurada, es aquella relación que contiene el mínimo de redundancia y permite a los usuarios insertar, modificar y borrar registros en una tabla sin errores o inconsistencias.

Anomalías en relaciones
Un error o inconsistencia que resulta cuando un usuario pretende actualizar una tabla que contiene datos redundantes. Tenemos tres tipos de anomalías, las cuales son: Anomalías en inserción, anomalías en eliminación, y anomalías en actualización.

Normalización
La normalización es un proceso formal para decidir que atributos deberían ser agrupados en una relación. La normalización es una herramienta para validar y mejorar el diseño lógico.

Normalización, es el proceso de descomponer relaciones con anomalías para producir relaciones pequeñas y bien estructuradas.

Una *forma normal* es un estado de una relación que resulta de aplicar simples reglas tomando en cuenta la dependencia funcional (o relaciones entre los atributos) de una relación.

Otra definición de Normalización sería:
“Proceso de eliminación de redundancias en una tabla para que sea más fácil de modificación"

4.6.1 Pasos en la Normalización

La normalización puede ser acompañada y entendida en etapas, cada una de ellas corresponde a una forma normal, estos pasos en la normalización son (Ver Figura 30):

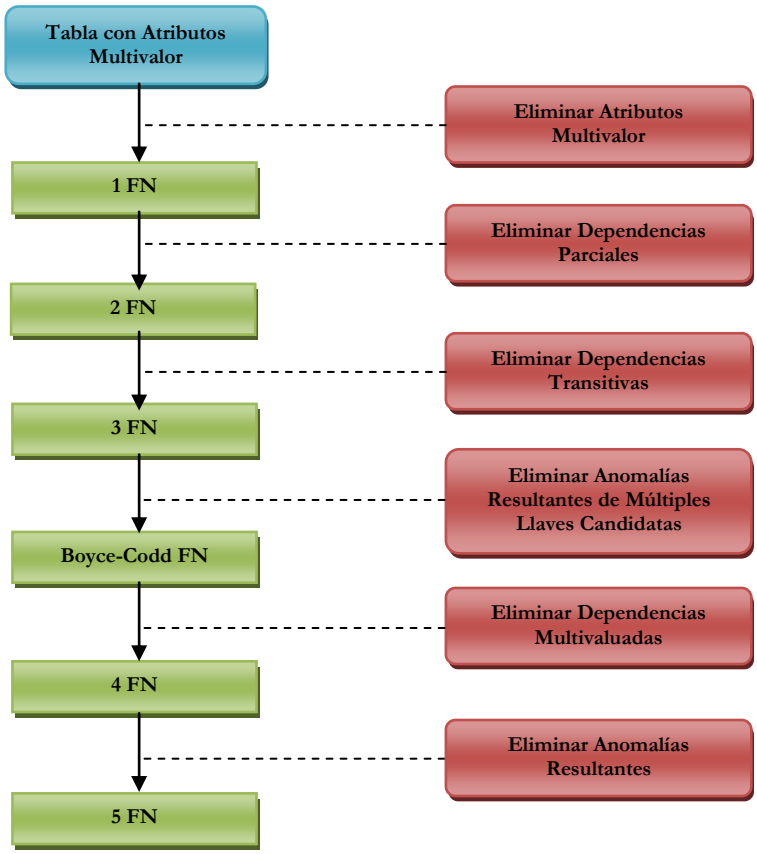


Figura 30. Pasos en el proceso de Normalización

1. Primera Forma Normal.
Cualquier atributo multivalor (también llamado grupo repetitivo) tiene que ser eliminado.

2. Segunda Forma Normal.
Cualquier dependendencia funcional parcial tienen que ser eliminadas, es decir, los atributos no claves son identificados por la llave primaria.

3. Tercera Forma Normal.
Cualquier dependendencia funcional parcial tienen que ser eliminadas, es decir, los atributos no claves son identificados por la llave primaria.

4. Boyce/Codd Forma Normal.
Cualquier anomalía resultante de dependendencias funcionales tienen que ser eliminadas.

5. Cuarta Forma Normal.
Cualquier dependendencia multivaluada tienen que ser eliminadas.

6. Quinta Forma Normal.
Cualquier dependendencia de junta o de proyección tienen que ser eliminadas.

4.6.2 Dependencias Funcionales

Uno de los conceptos fundamentales en la normalización es el de dependencia funcional.

Una *dependencia funcional* es una restricción entre dos conjuntos de atributos de la base de datos.

Supongamos que nuestra esquema de base de datos relacional tiene *n* atributos $A_1, A_2, ..., A_n$ y que toda la base de datos se describe con un sólo esquema de relación universal $R = A_1, A_2, ..., A_n$. Esto no implica que almacenaremos realmente la base de datos como una sola tabla universal; únicamente vamos a usar este concepto para desarrollar la teoría formal de las dependencias de datos.

Una dependencia funcional, denotada por $X \longrightarrow Y$, entre dos conjuntos de atributos X e Y que son subconjuntos de R, especifica una restricción sobre las posibles tuplas que podrán formar un estado de relación r de R.

La restricción dice que, para dos tuplas cualesquiera t_1 y t_2, de r tales que $t_1[X] = t_2[X]$, debemos tener también $t_1[Y] = t_2[Y]$. Esto significa que los valores del componente Y de una tupla r dependen de los valores del componente X, o están determinados por ellos; o bien, que los valores del componente X de una tupla determinan la manera única (o funcionalmente) los valores del componente Y.

También decimos que hay una dependencia funcional de X a Y o que Y depende funcionalmente de X.

Ejemplos:
1. NSS ⟶ Nombre, Dirección, Fecha de Nacimiento.
2. Placas ⟶ Marca, Modelo, Color
3. ISBN ⟶ Título, Nombre del primer Autor

El atributo que se encuentra del lado izquierdo de la flecha en una dependencia funcional es llamado *Determinante*.

4.6.3 Primera Forma Normal (1FN)

Establece que el dominio de un atributo debe incluir sólo *valores atómicos* (simples, indivisibles) y que el valor de cualquier atributo en una tupla debe ser un valor individual proveniente del dominio de ese atributo.

En otras palabras en la 1FN, debemos eliminar redundancia y convertir los atributos complejos en atributos atómicos, o no descomponibles (que ya no se pueden descomponer).

Considerando el esquema de la Figura 31, es evidente que la relación *Departamento* no está en 1FN porque *localizacionDepto* no es un atributo atómico, como puede verse en la primera tupla.

Departamento			
noDepto	**nombreDepto**	**nssJefeDepto**	**localizaciónDepto**
5	Investigación	3344555	Cuernavaca, México, Pachuca
4	Administración	9876543	Querétaro
1	Dirección	8886655	Cuernavaca

Figura 31. Relación *Departamento* sin normalizar

Al normalizar la relación quedaría como se ve la Figura 32, donde la definición de la llave primaria cambia, teniendo ahora una llave primaria compuesta por los atributos *noDepto,localizacionDepto*.

Departamento			
noDepto	**nombreDepto**	**nssJefeDepto**	**localizaciónDepto**
5	Investigación	3344555	Cuernavaca
5	Investigación	3344555	México
5	Investigación	3344555	Pachuca
4	Administración	9876543	Querétaro
1	Dirección	8886655	Cuernavaca

Figura 32. Relación *Departamento* en 1FN

4.6.4 Segunda Forma Normal (2FN)

Se basa el en concepto de *dependencia funcional total*. Una dependencia funcional $X \longrightarrow Y$ es una dependencia funcional total si la eliminación de cualquier atributo de A de X hace que la dependencia deje de ser válida; es decir, para cualquier atributo $A \in X$, $(X - \{A\})$ no determina funcionalmente a Y.

Una dependencia funcional $X \longrightarrow Y$ es una *dependencia parcial* si es posible eliminar un atributo A_X de X y la dependencia sigue siendo válida; es decir, para algún $A \in X$, $(X - \{A\}) \longrightarrow Y$.

Un esquema de relación R está en 2FN si todo atributo no primo A en R depende funcionalmente de manera total de la clave primaria de R.

En la 2FN, los atributos no clave depende total y funcionalmente de la llave primaria.

Considerando un ejemplo, en la Relación *EmpProyecto* (ver Figura 33):

EmpleadoProyecto					
nss	**noProyecto**	**horas**	**nombreEmpleado**	**nombreProyecto**	**localizaciónDepto**
123455	1	32.5	Carlos Pérez	Producto X	Cuernavaca
123455	2	7.5	Carlos Pérez	Producto Y	México
666884	3	40	Sonia González	Producto Z	Pachuca
453453	1	20	Israel Juárez	Producto X	Cuernavaca
453453	2	20	Israel Juárez	Producto Y	México
333454	2	10	Carmen Rodríguez	Producto Y	México
333454	3	10	Carmen Rodríguez	Producto Z	Pachuca
333454	10	10	Carmen Rodríguez	Automatización	Querétaro
333454	20	10	Carmen Rodríguez	Reorganización	Pachuca
892111	30	30	Alicia Guerrero	Prestación Nueva	Querétaro
892111	10	10	Alicia Guerrero	S	Querétaro
987155	10	35	Juan López	Automatización	Querétaro
987155	30	5	Juan López	S	Querétaro
987321	30	20	Roberto Suárez	Prestación Nueva	Querétaro
987321	20	15	Roberto Suárez	S	Pachuca
882255	20	3	Gerardo Hernández	Prestación Nueva	Pachuca

Figura 33. Relación *Emp_Proyecto* sin normalizar

{*nss, noProyecto*} ⟶ horas es una dependencia total (no se cumplen ni {*nss*}⟶ *horas* ni {*noProyecto*} ⟶ *horas*). Sin embargo la dependencia {*nss, noProyecto*}⟶ *nombreEmp* es parcial porque se cumple *nss* ⟶ *nombreEmp*.

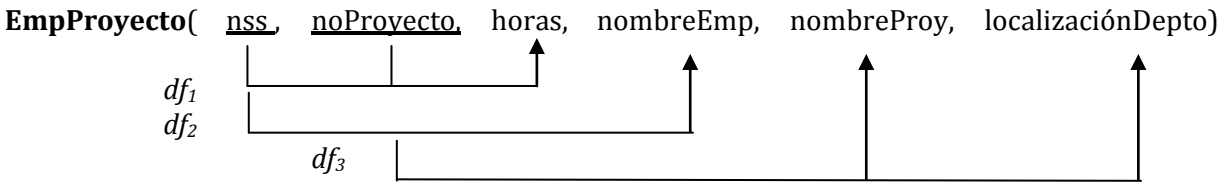


Figura 34. Dependencias Funcionales Parciales en la relación *Emp_Proyecto*

La prueba para 2FN incluye la verificación de dependencias funcionales cuyos atributos del miembro izquierdo son parte de la clave primaria. Si la llave primaria contiene un único atributo, no es en absoluto preciso aplicar la prueba.

La relación *EmpProyecto* está en 1FN pero no en 2FN (ver Figura 34). El atributo no primo *nombreEmp* viola 2FN debido a DF2, y lo mismo sucede con los atributos no primos *nombreProy* y *localizacionDepto* debido a DF3, las dependencias funcionales DF2 y DF3 hacen que *nombreEmp*, *nombreProy* y *localizacionDepto* dependan parcialmente de la clave primaria *nss*, *noProyecto* de *EmpProyecto*, violándose así la comprobación de 2FN.

Si un esquema de relación no está en 2FN, se le puede "normalizar en 2FN" dando lugar a varias relaciones 2FN 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. Así las dependencias funcionales DF1, DF2 y DF3 originan la descomposición de *EmpProyecto* en los tres esquemas de relación EP1, EP2 y EP3 que se ilustran a continuación, cada uno de los cuales está en 2FN.

Aplicando la 2FN la relación *EmpProyecto* queda con lo muestra la Figura 35.

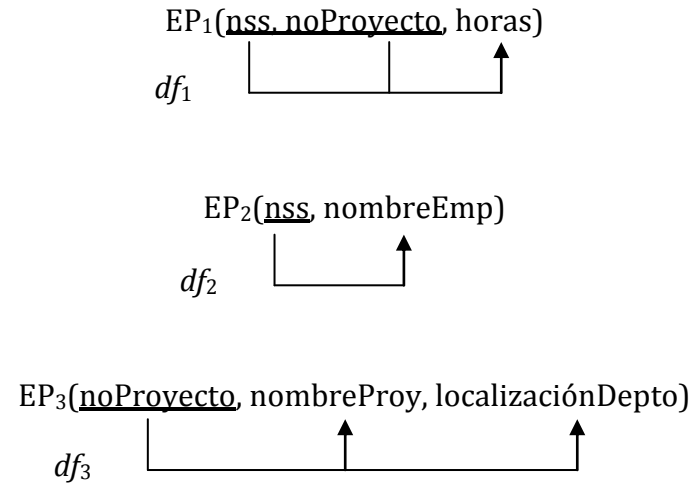


Figura 35. Relación *Emp_Proyecto* en 2FN

4.6.5 Tercera Forma Normal (3FN)

Se base en el concepto de *dependencia transitiva*. Una dependencia funcional $X \longrightarrow Y$ en un esquema de relación R es una *dependencia transitiva* si existe un conjunto de atributos Z que no sea subconjunto de cualquier clave R, y se cumplen tanto $X \longrightarrow Y$ como $Z \longrightarrow Y$, entonces $X \longrightarrow Z$.

De acuerdo con la definición original de Codd, un esquema de relación *R* está en 3FN si está en 2FN y ningún atributo no primo de *R depende transitivamente* de la clave primaria.

En la 3FN se elimina las dependencias transitivas; es decir atributos no clave no dependen de otros atributos no clave.

Considerando la relación *EmpDepartamento* (Ver Figura 36), la dependencia *nss⟶nssJefeDepto* es transitiva a través de *noDepto* de *EmpDepartamento*, porque se cumplen las dos dependencias *nss⟶noDepto* y *noDepto⟶nssJefeDepto* y *noDepto* no es ni una clave en sí misma ni un subconjunto de la clave de *EmpDepartamento*. Intuitivamente, podemos ver que en *EmpDepartamento* no es deseable la dependencia de *nssJefeDepto* con respecto a *noDepto* porque *noDepto* no es una clave de *EmpDepartamento*.

EmpDepartamento						
nombreEmpleado	**nss**	**fechaNacimiento**	**Dirección**	**noDepto**	**nombreDepto**	**nssJefeDepto**
Carlos Pérez	123455	09-Ene-1995	Av. J 3	5	Investigación	333454
Carmen Rodríguez	333454	08-Dic-1995	Fracc. L	5	Investigación	333454
Alicia Guerrero	987155	19-Sep-1998	Av. Hgo 4	4	Administración	987321
Juan López	987321	20-Jun-1991	Calle Gro	4	Administración	987321
Israel Juárez	666884	15-Sep-1992	Calle Fco	5	Investigación	333454
Sonia González	453453	31-Jul-1992	Fracc. O	5	Investigación	333454
Roberto Suárez	987155	29-May-1999	Blvd L. C	4	Administración	987321
Gerardo Hernández	882255	10-Nov-1997	Calle J	1	Dirección	882255

Figura 36. Relación *EmpDepartamento* sin normalizar

El esquema de relación *EmpDepartamento*, está en 2FN, pues no existen dependencias parciales sobre la clave. Sin embargo, no está en 3FN debido a que *nssJefeDepto* (y también *nombreDepto*) dependen transitivamente de *nss* a través de *noDepto* (Ver Figura 37).

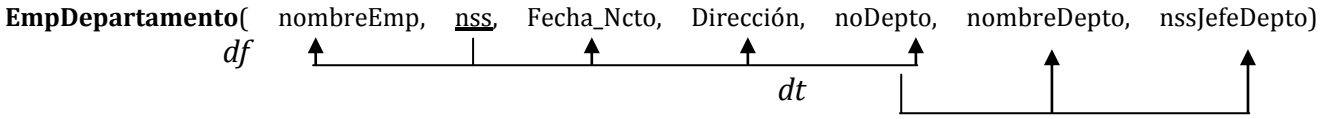


Figura 37. Dependencias Transitivas de la relación *EmpDepartamento*

Podemos normalizar *EmpDepartamento* descomponiéndolo en los dos esquemas de relación en 3FN, ED1 y ED2 que aparecen en la Figura 38. Intuitivamente, vemos que ED1 y ED2 representan hechos independientes acerca de las entidades empleados y departamentos.

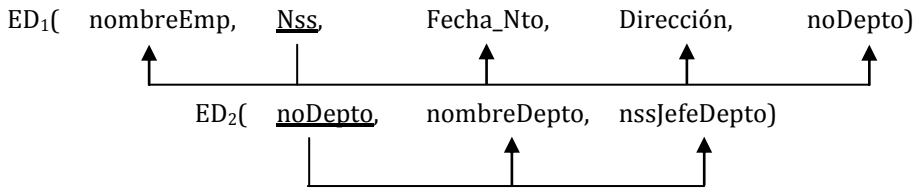


Figura 38. Relación *EmpDepartamento* en 3FN

4.6.6 Boyce/Codd Forma Normal (BCFN)

Cuando una relación tiene más de una llave candidata, anomalías pueden surgir a pesar de estar en 3FN. Considere el ejemplo de la relación *estudianteTutor* (Ver Figura 39), el cual tiene una llave compuesta por los atributos *noBoleta* y *área*. Los atributos *tutor* y *promedio* son totalmente dependientes de esta llave. Esto nos indica que para un determinado estudiante puede estar en más de un área, para cada área un estudiante tiene exactamente un *tutor* y un *promedio*.

Relación EstudianteTutor

noBoleta	**área**	**tutor**	**promedio**
123	Bases de datos	Idalia M.	6.0
123	Redes	Julia S.	7.2
456	Agentes de Software	Euler H.	5.3
789	Redes	Gilberto Q.	4.2
678	Bases de datos	Idalia M.	6.4

Figura 39. Relación *EstudianteTutor*

Existe una segunda dependencia funcional es esta relación: *área* es funcionalmente dependiente del *tutor*, esto implica que cada *tutor* es responsable exactamente de un *área* como se muestra en la Figura 40.

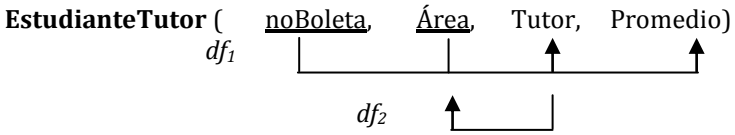


Figura 40. Dependencias funcionales en la Relación *EstudianteTutor*

Se definió anteriormente el concepto de dependencia transitiva como una dependencia funcional entre dos atributos no clave. En contraste en este ejemplo un atributo clave (*área*) es funcionalmente dependiente de un atributo no clave (*tutor*).
Anomalías en la relación *estudiantetutor*.

La relación *estudianteTutor* es claro que se encuentra en 3FN ya que no existen dependencias funcionales parciales y dependencias no transitivas. Sin embargo, por la existencia de la dependencia funcional entre *área* y *tutor*, existen anomalías en esta relación. Por ejemplo:

1. Supóngase que el tutor del área de BD es cambiado por Edgar C. Este cambio tendrá que aplicarse este cambio en dos registros (o más) en la tabla (*Anomalía en Actualización*).

2. Supóngase que queremos insertar la información de Ingeniería de Software cuyo tutor es Rosaura P. Esto no puede darse a cabo hasta que un estudiante de Ing de Sw sea asignado a Rosaura P. (*Anomalía en Inserción*).

3. Finalmente, si el estudiante con el noBoleta 789 deja la escuela, perderíamos la información de Gilberto Q. que es tutor de Redes. (*Anomalías de Eliminación*).

Forma Normal Boyce-Codd

"Una relación R está en la FNBC si y sólo si cada determinante es una llave candidata".

Las anomalías en la relación *estudianteTutor* son el resultado de la existencia del determinante tutor el cual no es una llave candidata en la relación *R*, por lo tanto

estudianteTutor no está en FNBC porque a pesar de que tutor es un determinante, este no es una llave candidata (solo el *área* es funcionalmente dependiente en *tutor*).

Convertir una relación en FNBC
Una relación que se encuentre en 3FN (pero no en FNBC) puede ser convertido a una relación en FNBC usando dos simples procesos.

La relación es modificada de tal manera que el determinante de la relación que no es una llave candidata llega ser un componente de la llave primaria de la relación revisada. El atributo que es funcionalmente dependiente en el determinante llega a ser un atributo no clave. Esto es válido por la dependencia funcional (Ver Figura 41).

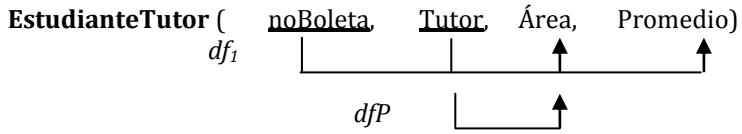


Figura 41. Relación *EstudianteTutor* en 2FN

Al examinar la relación *estudianteTutor* descubrirás que la nueva relación tiene una dependencia funcional parcial, al resolverlo se obtienen dos relaciones. Estas relaciones se encuentran en 3FN y también se encuentra en FNBC, ya que cada relación tiene una llave candidata (la llave primaria) (Ver Figura 41).

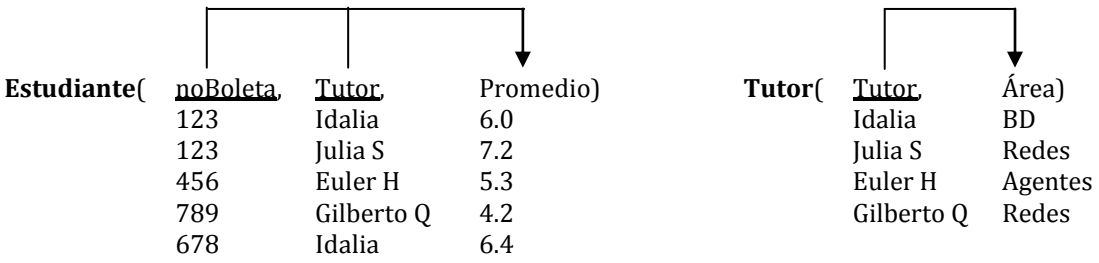


Figura 41. Relación *EstudianteTutor* en 3FN y FNBC