



INSTITUTO POLITÉCNICO NACIONAL



**Escuela Superior de Cómputo**

*Apuntes de:*

**BASE DE DATOS I**

**P R E S E N T A**

M. en C. Euler Hernández Contreras

México, D. F.

Octubre del 2008

# Contenido

<b>1</b>	<b>Modelo Relacional</b>	<b>1</b>
1.1	Introducción . . . . .	1
1.2	Conceptos Básicos . . . . .	1
1.2.1	Definiciones básicas . . . . .	1
1.2.2	Llaves Relacionales . . . . .	2
1.3	Propiedades de las relaciones . . . . .	2
1.4	Reglas de Integridad . . . . .	3
1.4.1	Nulo . . . . .	4
1.4.2	Regla de integridad de entidades . . . . .	4
1.4.3	Regla de integridad referencial . . . . .	4
1.4.4	Reglas del negocio . . . . .	5
1.5	Mapeo del DER al Modelo Relacional . . . . .	5
1.6	Normalización . . . . .	5
1.6.1	Pasos en la Normalización . . . . .	6
1.6.2	Dependencias Funcionales . . . . .	7
1.6.3	Primera Forma Normal (1FN) . . . . .	7
1.6.4	Segunda Forma Normal (2FN) . . . . .	8
1.6.5	Tercera Forma Normal (3FN) . . . . .	10
1.6.6	Boyce/Codd Forma Normal (BCFN) . . . . .	11

# Capítulo 1

## Modelo Relacional

### 1.1 Introducción

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

### 1.2 Conceptos Básicos

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

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

#### 1.2.1 Definiciones básicas

Algunas definiciones se comentan a continuación (Ver Figura 1.1):

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.

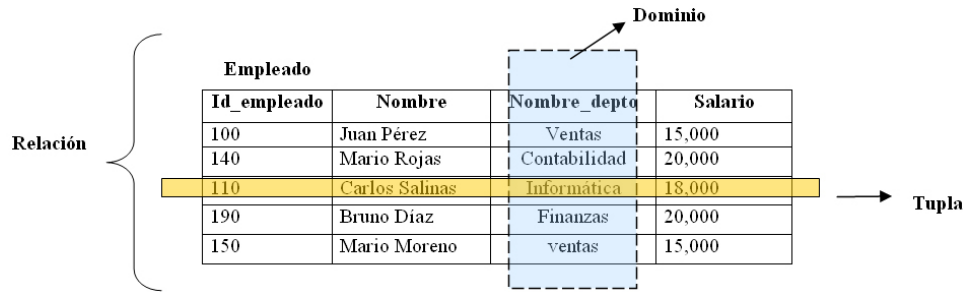


Figura 1.1: Algunas Definiciones Modelo Relacional

### 1.2.2 Llaves Relacionales

Una *llave primaria* 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 1.2)

Empleado			
<u>Id_empleado</u>	Nombre	Nombre_depto	Salario

Figura 1.2: Llave Primaria

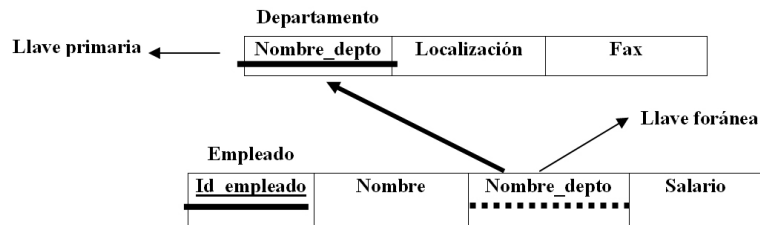
Una *llave compuesta* es una llave primaria que consiste de más de un atributo (Ver Figura 1.3).

Una *llave foránea* 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 1.4).

## 1.3 Propiedades de las relaciones

Las relaciones tienen las siguientes características:

Empleado			
<u>Id_empleado</u>	<u>Nombre_depto</u>	Nombre	Salario

Figura 1.3: *Llave Compuesta*Figura 1.4: *Llave Foránea*

- 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.

## 1.4 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:

- La regla de integridad de entidades.
- la regla de integridad referencial.

Antes de definir las, es preciso conocer el concepto de *nulo*.

### 1.4.1 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.

### 1.4.2 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. Que 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 base y a las claves primarias, no a las claves alternativas.

### 1.4.3 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.

### 1.4.4 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.

## 1.5 Mapeo del DER al Modelo Relacional

### 1.6 Normalización

#### *Relaciones bien estructuradas*

Antes de hablar de normalización, necesitamos preguntarnos, ¿cuando 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”*

### 1.6.1 Pasos en la Normalización

La normalización puede ser acompañada y entendida en etapas, cada una de ellas corresponde a un forma normal, estos pasos en la normalización son (Ver Figura 1.5):

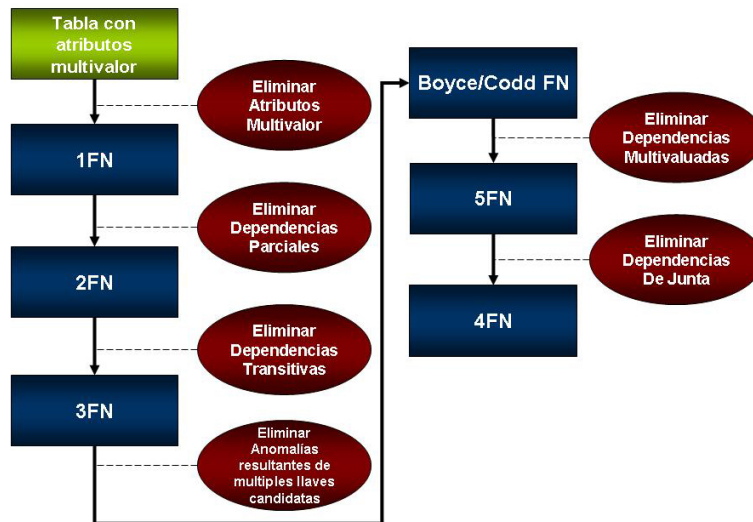


Figura 1.5: Pasos del proceso de Normalización

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.



### 1.6.2 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 nuestro esquema de base de datos relacional tiene  $n$  atributos  $A_1, A_2, \dots, A_n$  y que toda la base de datos se describe con un sólo esquema de relación universal  $R = A_1, A_2, \dots, A_n$ .

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 podrían 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  $\longrightarrow$  Nombre, Dirección, Fecha de Nacimiento.
2. Placas  $\longrightarrow$  Marca, Modelo, Color
3. ISBN  $\longrightarrow$  Título, Nombre del primer Autor

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

### 1.6.3 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 1.6, es evidente que la relación *Departamento* no está en 1FN porque *localizacionDepto* no es un atributo atómico, como puede verse en la primera tupla.

Al normalizar la relación quedaría como se ve la Figura 1.7, donde la definición de la llave primaria cambia, teniendo ahora una llave primaria compuesta por los atributos *noDepto, localizacionDepto*.

Departamento			
noDepto	nombreDepto	nssJefeDepto	localizacionDepto
5	Investigación	3344555	{Cuernavaca, México, Pachuca}
4	Administración	9876543	Querétaro
1	Dirección	8886655	Cuernavaca

Figura 1.6: *Relación Departamento sin Normalizar*

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

Figura 1.7: *Relación Departamento en la 1FN*

### 1.6.4 Segunda Forma Normal (2FN)

Se basa en el concepto de *dependencia funcional total*. Una dependencia funcional  $X \rightarrow Y$  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  $X \rightarrow Y$  es una *dependencia parcial* si es posible eliminar un atributo  $A \in X$  de  $X$  y la dependencia sigue siendo válida; es decir, para algún  $A \in X$ ,  $(X - \{A\}) \rightarrow Y$ .

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 dependen total y funcionalmente de la llave primaria.

Considerando un ejemplo, en la Relación *EmpProyecto* (ver Figura 1.8):

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

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 1.9). 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

Emp_Proyecto					
<u>nss</u>	<u>noProyecto</u>	horas	nombreEmp	nombreProy	localizacionDepto
123455	1	32.5	Pérez, Carlos	ProductoX	Cuernavaca
123455	2	7.5	Pérez, Carlos	ProductoY	México
666884	3	40	González, Sonia	ProductoZ	Pachuca
453453	1	20	Juárez, Israel	ProductoX	Cuernavaca
453453	2	20	Juárez, Israel	ProductoY	México
333454	2	10	Rodríguez, Carmen	ProductoY	México
333454	3	10	Rodríguez, Carmen	ProductoZ	Pachuca
333454	10	10	Rodríguez, Carmen	Automatización	Querétaro
333454	20	10	Rodríguez, Carmen	Reorganización	Pachuca
892111	30	30	Guerrero, Alicia	PrestacionNueva	Querétaro
892111	10	10	Guerrero, Alicia	S	Querétaro
987155	10	35	López Juan	Automatización	Querétaro
987155	30	5	López Juan	Automatización	Querétaro
987321	30	20	Suárez, Roberto	PrestaciónNueva	Querétaro
987321	20	15	Suárez, Roberto	S	Pachuca
882255	20	null	Hernández, Gerardo	PrestaciónNueva	Pachuca

Figura 1.8: Relación EmpProyecto sin Normalización

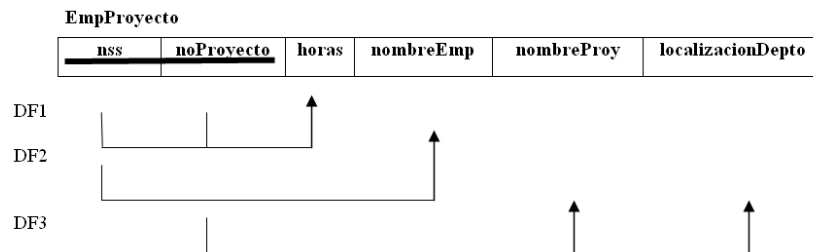
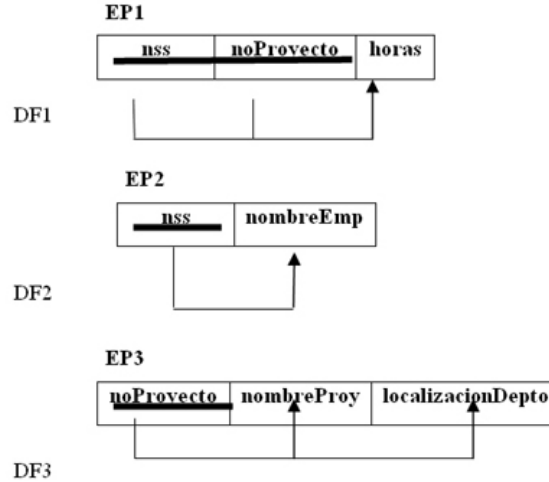


Figura 1.9: Dependencias Funcionales Parciales de la Relación EmpProyecto

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 1.10.

Figura 1.10: Relación *EmpProyecto* en 2FN

### 1.6.5 Tercera Forma Normal (3FN)

Se base en el concepto de *dependencia transitiva*. Una dependencia funcional  $X \rightarrow Y$  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  $X \rightarrow Z$  como  $Z \rightarrow Y$ , entonces  $X \rightarrow Y$ .

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.

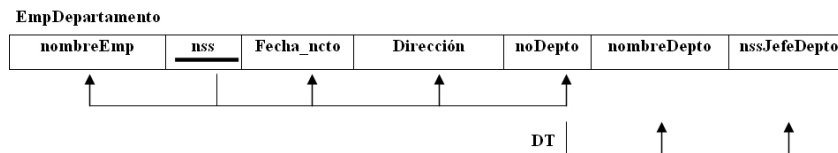
En la 3FN se eliminan las dependencias transitivas; es decir atributos no clave no dependen de otros atributos no clave.

Considerando la relación *EmpDepartamento* (Ver Figura 1.11), la dependencia  $nss \rightarrow nssJefeDepto$  es transitiva a través de noDepto de *EmpDepartamento*, porque se cumplen las dos dependencias  $nss \rightarrow noDepto$  y  $noDepto \rightarrow nssJefeDepto$  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*.

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 1.12). Podemos normalizar *EmpDepartamento* descomponiéndolo en los dos esquemas de relación en 3FN, ED1 y ED2 que aparecen en la Figura 1.13. Intuitivamente, vemos que ED1 y ED2 representan hechos independientes acerca de las entidades empleados y departamentos.

**EmpDepartamento**

nombreEmp	<u>nss</u>	Fecha_ncto	Dirección	noDepto	nombreDepto	nssJefeDepto
Pérez, Carlos	123455	09-01-1965	731 Av. Juárez 5	5	Investigación	333454
Rodríguez, Carmen	333454	08-12-1955	638 Fracc. Lomas	5	Investigación	333454
Guerrero, Alicia	987155	19-07-1968	3321 Av. Hidalgo	4	Administración	987321
López Juan	987321	20-06-1941	291 Calle Guerrero	4	Administración	987321
Juárez, Israel	666884	15-09-1962	975 Calle Fco. G.	5	Investigación	333454
González, Sonia	453453	31-07-1972	5631 Fracc. Oax	5	Investigación	333454
Suárez, Roberto	987155	29-03-1969	980 Blvd. Luis C.	4	Administración	987321
Hernández, Gerardo	882255	10-11-1937	450 Calle Juárez	1	Dirección	882255

Figura 1.11: Relación *EmpDepartamento* sin normalizarFigura 1.12: Dependencias Transitivas de la Relación *EmpDepartamento*

### 1.6.6 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 1.14), 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*.

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 1.15.

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 Nancy O. Este cambio

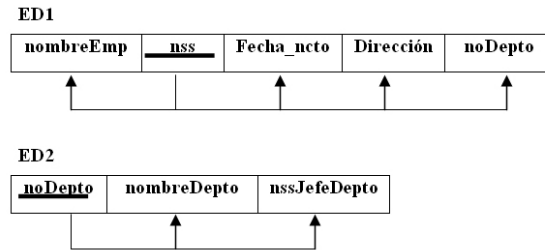


Figura 1.13: Relación EmpDepartamento en 3FN

estudianteTutor			
<u>noBoleta</u>	<u>área</u>	tutor	promedio
123	BD	Sergio S.	6.0
123	Redes	Julia S.	7.2
456	Agentes	Euler	5.3
789	Redes	Gilberto Q.	4.2
678	BD	Sergio S.	6.4

Figura 1.14: Relación estudianteTutor

tendrá que aplicarse este cambio en dos registros (o más) en la tabla (*Anomalía en Actualización*).

- 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*).
- 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*).

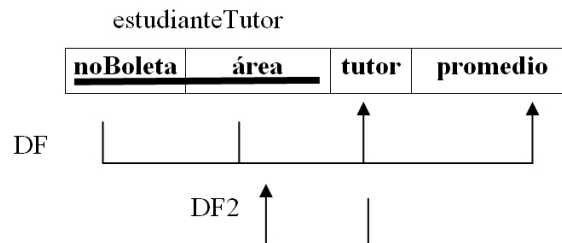
### Forma Normal Boyce-Codd

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

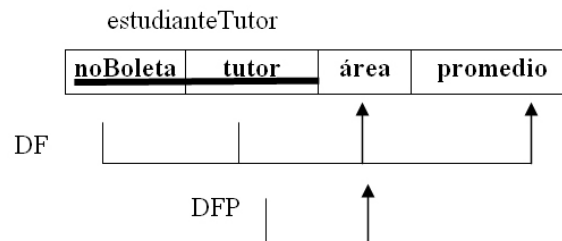
Las anomalías en la relación *estudianteTutor* son el resultado de la existencia del determinante *tutor* el cual no es una llave candidata en la relación  $R$ , por lo tanto *estudianteTutor* no está en FNBC porque a pesar de que *tutor* es un determinante, este no es una llave candidata (solo el *área* es funcionalmente dependiente en *tutor*).

### Convertir una relación en FNBC

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

Figura 1.15: *Relación estudianteTutor*

- La relación es modificada de tal manera que el determinante de la relación que no es una llave candidata llega a 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 1.16).

Figura 1.16: *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 1.17).

The diagram illustrates the decomposition of a table into two smaller tables. On the left, a table labeled 'estudiante' has three columns: 'noBoleta', 'tutor', and 'promedio'. Above it, a bracket indicates a decomposition into two tables. On the right, a table labeled 'Tutor' has two columns: 'tutor' and 'área'. Above it, a bracket indicates a decomposition into two tables. The data rows for each table are as follows:

estudiante		
noBoleta	tutor	promedio
123	Sergio S.	6.0
123	Julia S.	7.2
456	Euler	5.3
789	Gilberto Q.	4.2
678	Sergio S.	6.4

Tutor	
tutor	área
Sergio S.	BD
Julia S.	Redes
Euler	Agentes
Gilberto Q.	Redes

Figura 1.17: Relación *estudianteTutor* en 3FN y FNBC