

UNIDAD III - MODELO RELACIONAL

PARTE II - NORMALIZACIÓN

BASES DE DATOS



I. NORMALIZACIÓN

El diseño de una BD relacional se puede realizar aplicando al mundo real, en una primera fase, un modelo como el modelo E/R, a fin de obtener un esquema conceptual; en una segunda fase, se transforma dicho esquema al modelo relacional mediante las correspondientes **reglas de transformación**.

También es posible, aunque quizás menos recomendable, obtener el esquema relacional sin realizar ese paso intermedio que es el esquema conceptual.

En ambos casos, es conveniente (obligatorio en el modelo relacional directo) aplicar un conjunto de reglas, conocidas como **Teoría de normalización**, que nos permiten asegurar que un esquema relacional cumple unas ciertas propiedades

I. NORMALIZACIÓN

Con la **Teoría de normalización** pretendemos evitar:

La redundancia de los datos: repetición de datos en un sistema.

- Anomalías de actualización: inconsistencias de los datos como resultado de datos redundantes y actualizaciones parciales.
- Anomalías de borrado: pérdidas no intencionadas de datos debido a que se han borrado otros datos.
- Anomalías de inserción: imposibilidad de adicionar datos en la base de datos debido a la ausencia de otros datos.

En la práctica, si la BD se ha diseñado haciendo uso de modelos semánticos como el modelo E/R no suele ser necesaria la normalización. Por otro lado si nos proporcionan una base de datos creada sin realizar un diseño previo, es muy probable que necesitemos normalizar.

I. NORMALIZACIÓN

En la teoría de bases de datos relacionales, **las formas normales (FN) proporcionan los criterios para determinar el grado de vulnerabilidad de una tabla a inconsistencias y anomalías lógicas.**

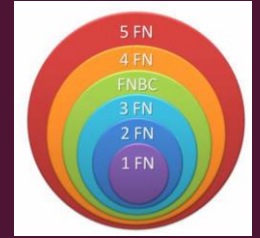
Cuanto más alta sea la forma normal aplicable a una tabla, menos vulnerable será a inconsistencias y anomalías.

- **Edgar F. Codd originalmente definió las tres primeras** formas normales (1FN, 2FN, y 3FN) en 1970. Estas formas normales se han resumido como requiriendo que todos los atributos sean atómicos, dependan de la clave completa y en forma directa (no transitiva).
- La forma normal de **Boyce-Codd (FNBC)** fue introducida en 1974 por los dos autores que aparecen en su denominación.
- Las cuarta y quinta formas normales (4FN y 5FN) se ocupan específicamente de la representación de las relaciones muchos a muchos y uno a muchos entre los atributos y fueron introducidas por **Fagin** en 1977 y 1979 respectivamente.

Cada forma normal incluye a las anteriores.

I. NORMALIZACIÓN

DEFINICIONES



➤ **Dependencia funcional:** $A \rightarrow B$, representa que B es funcionalmente dependiente de A. Para **un valor de A** siempre aparece **un valor de B**.

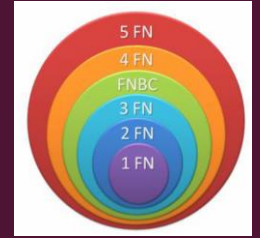
Ejemplo: Si A es el D.N.I., y B el Nombre, está claro que para un número de D.N.I., siempre aparece el mismo nombre de titular.

Las dependencias funcionales determinan una manera de definir restricciones en un esquema relacional.

Dada una relación R que contiene los atributos X e Y se dice que Y depende funcionalmente de X ($X \rightarrow Y$) sí y sólo sí en todo momento cada valor de X tiene asociado un solo valor de Y. Esto es lo mismo que decir que si dos tuplas de R tienen el mismo valor para su atributo X forzosamente han de tener el mismo valor para el atributo Y.

I. NORMALIZACIÓN

DEFINICIONES

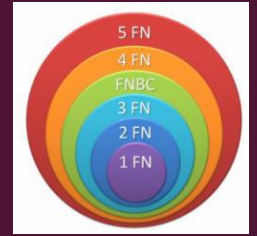


- **Dependencia funcional completa:** $A \rightarrow B$, si B depende de A en su totalidad.
Ejemplo: Tiene sentido plantearse este tipo de dependencia cuando A está compuesto por más de un atributo.

Por ejemplo, supongamos que A corresponde al atributo compuesto: D.N.I._Empleado + Cod._Dpto. y B es Nombre_Dpto. En este caso B depende del Cod_Dpto., pero no del D.N.I._Empleado. Por tanto en este ejemplo no habría dependencia funcional completa.

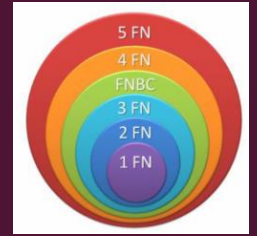
I. NORMALIZACIÓN

DEFINICIONES



- **Dependencia transitiva:** $A \rightarrow B \rightarrow C$. Si $A \rightarrow B$ y $B \rightarrow C$, Entonces decimos que C depende de forma transitiva de A. Ejemplo: Sea A el D.N.I. de un alumno, B la localidad en la que vive y C la provincia. Es un caso de dependencia transitiva $A \rightarrow B \rightarrow C$.
- **Determinante funcional:** todo atributo, o conjunto de ellos, de los que depende algún otro atributo. Ejemplo: El D.N.I. es un **determinante** funcional pues atributos como nombre, dirección, localidad, etc, dependen de él.
- **Dependencia multivaluada:** $A \twoheadrightarrow B$. Son un tipo de dependencias en las que un determinante funcional no implica un único valor, sino un conjunto de ellos. **Un valor de A** siempre implica **varios valores de B**. Ejemplo: CursoBachillerato \twoheadrightarrow Modalidad. Para primer curso siempre va a aparecer en el campo Modalidad uno de los siguientes valores: Ciencias, Humanidades/Ciencias Sociales o Artes. Igual para segundo curso.

I. NORMALIZACIÓN PROCESO.



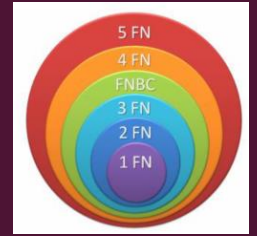
El **proceso de normalización** consiste en seguir una serie de pasos o normas, al definir una Base de Datos Relacional que, tras ser aplicadas, se obtienen los datos agrupados en diferentes relaciones, de forma tal que la estructura obtenida es óptima para su implementación, gestión y aplicación desde diferentes aplicaciones futuras.

Se dice que una relación está en una forma normal cuando satisface un conjunto de restricciones impuestas por dicha norma.

La normalización se basa en que “los datos son independientes de las aplicaciones que los gestionan” y su objetivo es “obtener el mayor número de relaciones posible, dejando en cada una de ellas los atributos imprescindibles para representar a la entidad (objeto) o a la relación entre entidades a la que hace referencia la relación mediante la conexión de sus claves”.

I. NORMALIZACIÓN

VENTAJAS.



Las ventajas que se obtienen tras la normalización de los datos para su **eficaz gestión** son:

- **Facilidad de uso.** - Los datos están agrupados en relaciones que identifican claramente un objeto o una relación.
- **Flexibilidad.** - La información que necesitan los usuarios puede obtenerse de las relaciones relacionales o las relaciones mediante operaciones de álgebra relacional, uniendo relaciones, seleccionando sus valores, proyectándolos, etc.
- **Precisión.** - Las interrelaciones entre relaciones consiguen mantener información diferente relacionada con toda exactitud.
- **Seguridad.** - Los controles de acceso para consultar o actualizar información (tanto en relaciones como en atributos) son mucho más sencillos de implementar.
- **Independencia de datos.** - Los programas no están ligados a las estructuras, con lo que se consigue aumentar la base de datos añadiendo nuevos atributos o nuevas relaciones sin que afecten a los programas que las usan.
- **Claridad.** - La representación de la información es clara y sencilla para un usuario: son relaciones simples.
- **Facilidad de gestión.** - Los lenguajes manipulan la información en forma sencilla, al estar los datos basados en el álgebra y cálculo relacional.
- **Mínima redundancia.** - La información no estará duplicada innecesariamente dentro de las estructuras.
- **Máximo rendimiento** de las aplicaciones. - Sólo se trata aquella información que va a ser de utilidad en cada aplicación concreta.

II. PRIMERA FORMA NORMAL. 1FN

Una relación está en 1FN si y sólo si **cada atributo es atómico**. Esto es: Se dice que una relación está en 1FN si y sólo si cada atributo de la relación toma un único valor del dominio correspondiente, es decir, si y sólo si cada atributo depende funcionalmente de la clave principal

Ejemplo: Supongamos que tenemos la siguiente tabla con datos de alumnado de un centro de enseñanza secundaria

DNI	Nombre	Curso	FechaMatrícula	Tutor	LocalidadAlumno	ProvinciaAlumno	Teléfonos
11111111A	Eva	1ESO-A	01-Julio-2016	Isabel	Écija	Sevilla	660111222
22222222B	Ana	1ESO-A	09-Julio-2016	Isabel	Écija	Sevilla	660222333 660333444 660444555
33333333C	Susana	1ESO-B	11-Julio-2016	Roberto	Écija	Sevilla	
44444444D	Juan	2ESO-A	05-Julio-2016	Federico	El Villar	Córdoba	
55555555E	José	2ESO-A	02-Julio-2016	Federico	El Villar	Córdoba	661000111 661000222



¿Está en 1FN? ¿Qué soluciones propones?

II. PRIMERA FORMA NORMAL. 1FN

Como se puede observar, esta tabla no está en 1FN puesto que el campo Teléfonos contiene varios datos dentro de una misma celda y por tanto no es un campo cuyos valores sean atómicos.

Posibles soluciones:

1) Añadir tantos atributos como valores pueda tomar el atributo no atómico. Aplicable sólo cuando conocemos con certeza los valores que puede tomar el atributo no atómico.

R (cr1, cr2, cr31, cr32, cr33, cr4)

ALUMNOS (dni, nombre, curso, fecha_matriculatfijo, tutor, localidad, provincia, teléfono_móvil, teléfono_fijo)

2) Descomponer la relación (**MEJOR SOLUCIÓN**):

La relación o tabla R se descompone en 2 relaciones R1 y R2 de la siguiente manera:

R1 (cr1, cr2, cr4) Se quedaría con la clave principal y los atributos que dependen funcionalmente de ésta.

II. PRIMERA FORMA NORMAL. IFN

La solución quedaría como :

Alumnos

DNI	Nombre	Curso	FechaMatrícula	Tutor	LocalidadAlumno	ProvinciaAlumno
11111111A	Eva	1ESO-A	01-Julio-2016	Isabel	Écija	Sevilla
22222222B	Ana	1ESO-A	09-Julio-2016	Isabel	Écija	Sevilla
33333333C	Susana	1ESO-B	11-Julio-2016	Roberto	Écija	Sevilla
44444444D	Juan	2ESO-A	05-Julio-2016	Federico	El Villar	Córdoba
55555555E	José	2ESO-A	02-Julio-2016	Federico	El Villar	Córdoba

Teléfonos

DNI	Teléfono
11111111A	660111222
22222222B	660222333
22222222B	660333444
22222222B	660444555
55555555E	661000111
55555555E	661000222

II. PRIMERA FORMA NORMAL. 1FN

Otro ejemplo, si tengo esta tabla, ¿cuáles serían las posibilidades de descomposición?

COD_EMP	NOMBRE	IDIOMA
5050	M. DORADO	ITALIANO ESPAÑOL
6969	P. SUAREZ	ESPAÑOL
...

A)

COD_EMP	NOMBRE	IDIOMA
5050	M. DORADO	ITALIANO
5050	M. DORADO	ESPAÑOL
6969	P. SUAREZ	ESPAÑOL
...

B)

COD_EMP	NOMBRE
5050	M. DORADO
6969	P. SUAREZ
...	...

COD_EMP	IDIOMA
5050	ITALIANO
5050	ESPAÑOL
6969	ESPAÑOL
...	...

C)

COD_EMP	NOMBRE	IDIOMA1	IDIOMA2	...
5050	M. DORADO	ESPAÑOL	ITALIANO	...
6969	P. SUAREZ	ESPAÑOL		...
...

¿Cuál sería la mejor opción?

II. PRIMERA FORMA NORMAL. 1FN

Ejemplo 3:

Tabla

CLIENTE (dni, nombre, teléfonos, población)

Valores:

CLIENTE (11111, pepe, 914569248 - 654678594, Madrid)



¿Cuáles serían las soluciones?

II. PRIMERA FORMA NORMAL. 1FN

Ejemplo 3:

Tabla

CLIENTE (dni, nombre, teléfonos, población)

Valores:

CLIENTE (11111, pepe, 914569248 - 654678594, Madrid)

R (cr1, cr2, cr3, cr4) {cr1 → cr2, cr1 → cr4} cr1 ↗ cr3

cr3 no atómico, puede tomar 3 valores



¿Cuáles serían las soluciones?

II. PRIMERA FORMA NORMAL. 1FN

OPCIÓN 1: Añadir tantos atributos como valores pueda tomar el atributo no atómico.

Aplicable sólo cuando **conocemos con certeza los valores que puede tomar el atributo no atómico.**

R (cr1, cr2, cr31, cr32, cr33, cr4)

CLIENTE (dni, nombre, tfijo, tmóvil, población)

OPCIÓN 2: Descomponer la relación (**MEJOR SOLUCIÓN**):

La relación o tabla R se descompone en 2 relaciones R1 y R2 de la siguiente manera:

R1 (cr1, cr2, cr4) Se quedaría con la clave principal y los atributos que dependen funcionalmente de ésta.

CLIENTE (dni, nombre, población)

R2 (cr1, cr3) donde cr1 clave ajena. R2 estaría formada por:

- la clave principal de R formando parte de la clave principal de R2 y como clave ajena de R1
- atributos que no dependían funcionalmente de la clave principal de R.

TELEFONOS (cliente, telefono)

II. PRIMERA FORMA NORMAL. 1FN

Si hubiera más atributos no dependientes de la clave principal de R y que formen parte de R2, tenemos que comprobar que dependen funcionalmente de la clave principal de R2.

Ejemplo:

PRODUCTOS (CODPROD, NOMPROD, CODALM, NOMALM, TELALM, cantidad) teniendo en cuenta que **un mismo producto puede estar almacenado en diversos almacenes**, en cada uno de ellos una determinada cantidad {CODPROD, CODALM \rightarrow cantidad}
{CODPROD \rightarrow NOMPROD, CODALM \rightarrow NOMALM, TELALM}

por tanto CODPRO \twoheadrightarrow CODALM, NOMALM, TELALM

OPCIÓN descomposición:

PRODUCTOS (CODPRO, NOMPROD) 1FN

PRODUCTOS_ALMACEN (CODPRO, CODALM, NOMALM, TELALM, cantidad) 1FN porque se cumple
CODPRO, CODALM \rightarrow NOMALM, TELALM, cantidad

II. SEGUNDA FORMA NORMAL. 2FN

Una Relación esta en 2FN si y sólo si está en 1FN y todos los atributos que no forman parte de la Clave Principal tienen dependencia funcional completa de ella.

Es decir, se dice una relación se encuentra en 2FN si y sólo si cumple las condiciones siguientes:

- Se encuentra en la Primera Forma Normal.
- Todo atributo secundario (es decir, todo atributo de la relación que no pertenece a la clave principal) tiene una **dependencia funcional completa** de la clave principal.

Dado que la clave principal puede ser una clave compuesta, una relación no estará en la 2FN si algún atributo de ella depende funcionalmente de una parte de la clave principal pero no de la clave completa

II. SEGUNDA FORMA NORMAL. 2FN

Ejemplo: Seguimos con el ejemplo anterior. Trabajaremos con la siguiente tabla:

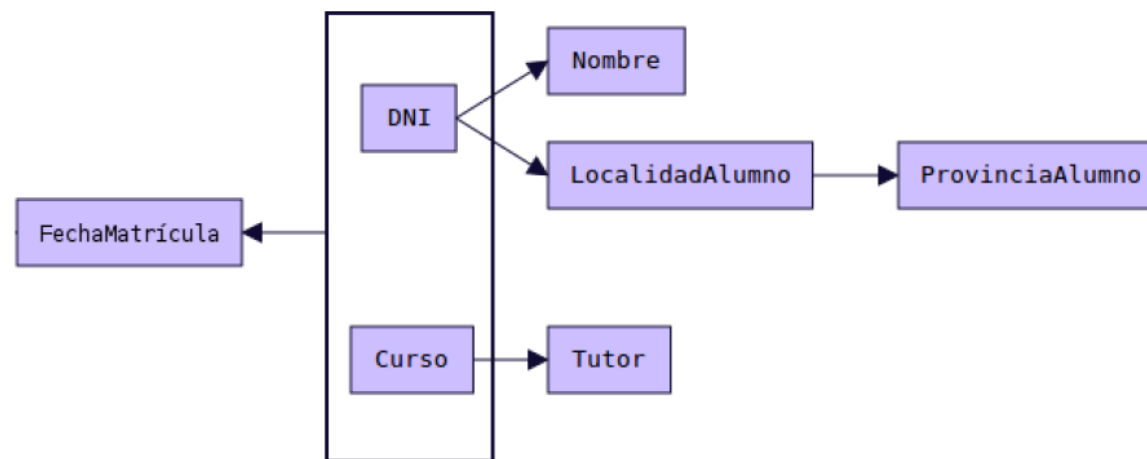
Alumnos						
DNI	Nombre	Curso	FechaMatrícula	Tutor	LocalidadAlumno	ProvinciaAlumno
11111111A	Eva	1ESO-A	01-Julio-2016	Isabel	Écija	Sevilla
22222222B	Ana	1ESO-A	09-Julio-2016	Isabel	Écija	Sevilla
33333333C	Susana	1ESO-B	11-Julio-2016	Roberto	Écija	Sevilla
44444444D	Juan	2ESO-A	05-Julio-2016	Federico	El Villar	Córdoba
55555555E	José	2ESO-A	02-Julio-2016	Federico	El Villar	Córdoba



¿Está en 2FN? ¿Qué soluciones propones?

II. SEGUNDA FORMA NORMAL. 2FN

Vamos a examinar las dependencias funcionales. El gráfico que las representa :



- Siempre que aparece un DNI aparecerá el Nombre correspondiente y la LocalidadAlumno correspondiente. Por tanto $DNI \rightarrow Nombre$ y $DNI \rightarrow LocalidadAlumno$. Por otro lado siempre que aparece un Curso aparecerá el Tutor correspondiente. Por tanto $Curso \rightarrow Tutor$. Los atributos Nombre y LocalidadAlumno no dependen funcionalmente de Curso, y el atributo Tutor no depende funcionalmente de DNI.
- El único atributo que sí depende de forma completa de la clave compuesta DNI y Curso es FechaMatrícula: $(DNI, Curso) \rightarrow FechaMatrícula$.

II. SEGUNDA FORMA NORMAL. 2FN

A la hora de establecer la **Clave Primaria** de una tabla debemos escoger un atributo o conjunto de ellos de los que dependan funcionalmente el resto de atributos.

Además debe ser una **dependencia funcional completa**.

Si escogemos DNI como clave primaria, tenemos un atributo (Tutor) que no depende funcionalmente de él. Si escogemos Curso como clave primaria, tenemos otros atributos que no dependen de él. Si escogemos la combinación (**DNI, Curso**) como clave primaria, entonces sí tenemos todo el resto de atributos con dependencia funcional respecto a esta clave. Pero es una **dependencia parcial**, no total (salvo FechaMatrícula, donde sí existe dependencia completa).

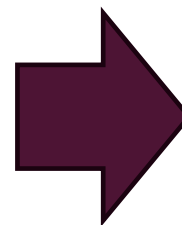
Por tanto, **esta tabla no está en 2FN**.

¿Cuál sería la solución?

II. SEGUNDA FORMA NORMAL. 2FN

Alumnos

DNI	Nombre	Curso	FechaMatrícula	Tutor	LocalidadAlumno	ProvinciaAlumno
11111111A	Eva	1ESO-A	01-Julio-2016	Isabel	Écija	Sevilla
22222222B	Ana	1ESO-A	09-Julio-2016	Isabel	Écija	Sevilla
33333333C	Susana	1ESO-B	11-Julio-2016	Roberto	Écija	Sevilla
44444444D	Juan	2ESO-A	05-Julio-2016	Federico	El Villar	Córdoba
55555555E	José	2ESO-A	02-Julio-2016	Federico	El Villar	Córdoba



Alumnos

DNI	Nombre	Localidad	Provincia
11111111A	Eva	Écija	Sevilla
22222222B	Ana	Écija	Sevilla
33333333C	Susana	El Villar	Córdoba
44444444D	Juan	El Villar	Córdoba
55555555E	José	Écija	Sevilla

Matrículas

DNI	Curso	FechaMatrícula
11111111A	1ESO-A	01-Julio-2016
22222222B	1ESO-A	09-Julio-2016
33333333C	1ESO-B	11-Julio-2016
44444444D	2ESO-A	05-Julio-2016
55555555E	2ESO-A	02-Julio-2016

Cursos

Curso	Tutor
1ESO-A	Isabel
1ESO-B	Roberto
2ESO-A	Federico

II. SEGUNDA FORMA NORMAL. 2FN

TEOREMA DE LA SEGUNDA FORMA NORMAL

Sea una Relación formada por los atributos A, B, C, D con clave primaria compuesta por los atributos A y B.

Si se cumple que: $A \rightarrow D$

Entonces la Relación puede descomponerse en dos Relaciones Relación1 y Relación2 con los atributos respectivos:

RELACIÓN1 (A, D)

RELACIÓN2 (A, B, C)

Si una relación no está en 2FN:

S (cs1, cs2, cs3, cs4, cs5) se da $cs2 \rightarrow cs5$ (hay que sacar fuera de la relación estos atributos,)

$cs1, cs2 \rightarrow cs5$ es DF no completa

Descomponer:

S1 (cs1, cs2, cs3, cs4)

S2 (cs2, cs5) Atributos que hacen que la DF no sea completa

II. SEGUNDA FORMA NORMAL. 2FN

TEOREMA DE LA SEGUNDA FORMA NORMAL

Ejemplo:

DETALLEVIAJE (codviaje, codconductor, puesto, nombre, teléfono) está en 1FN

{codviaje, codconductor → puesto
codviaje, codconductor → nombre
codviaje, codconductor → teléfono
Codconductor → nombre, teléfono}

codconductor → nombre esta DF hace que codviaje, codconductor → nombre no sea DF completa
codconductor → telefono esta DF hace que codviaje, codconductor → teléfono no es DF completa

Descomponer en DETALLEVIAJE y CONDUCTOR

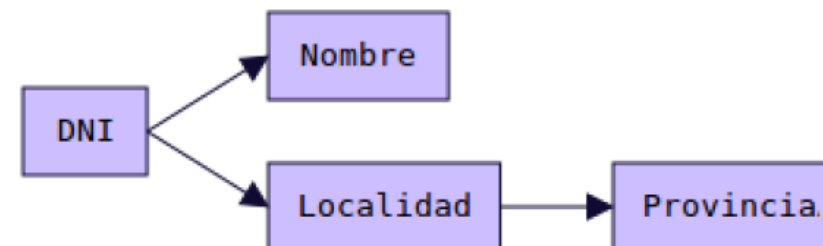
DETALLEVIAJE (codviaje, codconductor, puesto) está en 2FN
CONDUCTOR (codconductor, nombre, teléfono) está en 2FN

III. TERCERA FORMA NORMAL. 3FN

Una Relación esta en 3FN si y sólo si está en 2FN y **no existen dependencias transitivas**. Todas las dependencias funcionales deben ser respecto a la clave principal. Ejemplo: Seguimos con el ejemplo anterior. Trabajaremos con la siguiente tabla:

Alumnos

DNI	Nombre	Localidad	Provincia
11111111A	Eva	Écija	Sevilla
22222222B	Ana	Écija	Sevilla
33333333C	Susana	El Villar	Córdoba
44444444D	Juan	El Villar	Córdoba
55555555E	José	Écija	Sevilla



Las dependencias funcionales existentes son las siguientes. Como podemos observar existe una dependencia funcional transitiva: $DNI \rightarrow Localidad \rightarrow Provincia$

III. TERCERA FORMA NORMAL. 3FN

Para que la tabla esté en 3FN, **no pueden existir dependencias funcionales transitivas**. Para solucionar el problema deberemos crear una nueva tabla. El resultado es:

Alumnos

DNI	Nombre	Localidad
11111111A	Eva	Écija
22222222B	Ana	Écija
33333333C	Susana	El Villar
44444444D	Juan	El Villar
55555555E	José	Écija

Localidades

Localidad	Provincia
Écija	Sevilla
El Villar	Córdoba

Jura usted que cada columna de cada fila **depende**:
De la clave (1FN)
De toda la clave (2FN) y
Nada mas que de la clave (3FN)



Alumnos

DNI	Nombre	Localidad
11111111A	Eva	Écija
22222222B	Ana	Écija
33333333C	Susana	El Villar
44444444D	Juan	El Villar
55555555E	José	Écija

Teléfonos

DNI	Teléfono
11111111A	660111222
22222222B	660222333
22222222B	660333444
22222222B	660444555
55555555E	661000111
55555555E	661000222

Cursos

Curso	Tutor
1ESO-A	Isabel
1ESO-B	Roberto
2ESO-A	Federico

Localidades

Localidad	Provincia
Écija	Sevilla
El Villar	Córdoba

Matrículas

DNI	Curso	FechaMatrícula
11111111A	1ESO-A	01-Julio-2016
22222222B	1ESO-A	09-Julio-2016
33333333C	1ESO-B	11-Julio-2016
44444444D	2ESO-A	05-Julio-2016
55555555E	2ESO-A	02-Julio-2016

III. TERCERA FORMA NORMAL. 3FN

Es decir, se dice que una relación está en 3FN sí y sólo si cumple las siguientes condiciones:

- Se encuentra en la Segunda Forma Normal.
- Ningún atributo no primario es transitivamente dependiente de cada posible clave (primaria o candidatas) de la relación.

Esto quiere decir que no existe **ningún atributo no principal que dependa transitivamente de alguna de las claves de la relación.**

R (cr1, cr2, cr3, cr4, cr5)

$cr1 \rightarrow cr3 \rightarrow cr5$

$cr3 \twoheadrightarrow cr1$

Descomponer:

R1 (cr1, cr2, cr3, cr4)

R2 (cr3, cr5) atributos que dependen transitivamente de la clave principal

III. TERCERA FORMA NORMAL. 3FN

Ejemplo:

EMPLEADO (codempleado, nombre, dirección, población, teléfono, coddpto, nomdpto, ubicacióndpto)

Codempleado \rightarrow coddpto \rightarrow nomdpto, ubicacióndpto

Coddpto \mapsto codempleado

Descomponer en:

EMPLEADO (codempleado, nombre, dirección, población, teléfono, dpto) está 3FN

DPTO (coddpto, nomdpto, ubicacióndpto) está en 3FN

IV. FORMA NORMAL DE BOYCE-CODD: FNBC

Una Relación esta en FNBC si está en 3FN y no existe solapamiento de claves candidatas. Solamente hemos de tener en cuenta esta forma **normal cuando tenemos varias claves candidatas compuestas y existe solapamiento entre ellas.**

Pocas veces se da este caso.

Ejemplo: Tenemos una tabla con información de proveedores, códigos de piezas y cantidades de esa pieza que proporcionan los proveedores. Cada proveedor tiene un nombre único. Los datos son:

Suministros			
CIF	Nombre	CódigoPieza	CantidadPiezas
S-11111111A	Ferroman	1	10
B-22222222B	Ferrotex	1	7
M-33333333C	Ferropet	3	4
S-11111111A	Ferroman	2	20
S-11111111A	Ferroman	3	15
B-22222222B	Ferrotex	2	8
B-22222222B	Ferrotex	3	4

IV. FORMA NORMAL DE BOYCE-CODD: FNBC

La solución sería:

Proveedores

CIF	Nombre
S-11111111A	Ferroman
B-22222222B	Ferrotex
M-33333333C	Ferropet

Suministros

CIF	CódigoPieza	CantidadPiezas
S-11111111A	1	10
B-22222222B	1	7
M-33333333C	3	4
S-11111111A	2	20
S-11111111A	3	15
B-22222222B	2	8
B-22222222B	3	4

V. CUARTA FORMA NORMAL: 4FN

Una Relación esta en 4FN si y sólo si está en 3FN (o FNBC) y las únicas dependencias multivaluadas son aquellas que dependen de las claves candidatas.

Ejemplo: Tenemos una tabla con la información de nuestros alumnos y alumnas y las asignaturas que cursan así como los deportes que practican

Alumnado

Estudiante	Asignatura	Deporte
11111111A	Matemáticas, Lengua	Natación, Baloncesto
22222222B	Matemáticas	Fútbol, Natación

Alumnado

Estudiante	Asignatura	Deporte
11111111A	Matemáticas	Natación
11111111A	Matemáticas	Baloncesto
11111111A	Lengua	Natación
11111111A	Lengua	Baloncesto
22222222B	Matemáticas	Fútbol
22222222B	Matemáticas	Natación

V. CUARTA FORMA NORMAL: 4FN

Para normalizar esta tabla, debemos darnos cuenta que la oferta de asignaturas está compuesta por un conjunto de valores limitado. Igual sucede con los deportes. Por tanto existen dos dependencias multivaluadas:

Estudiante \twoheadrightarrow Asignatura

Estudiante \twoheadrightarrow Deporte

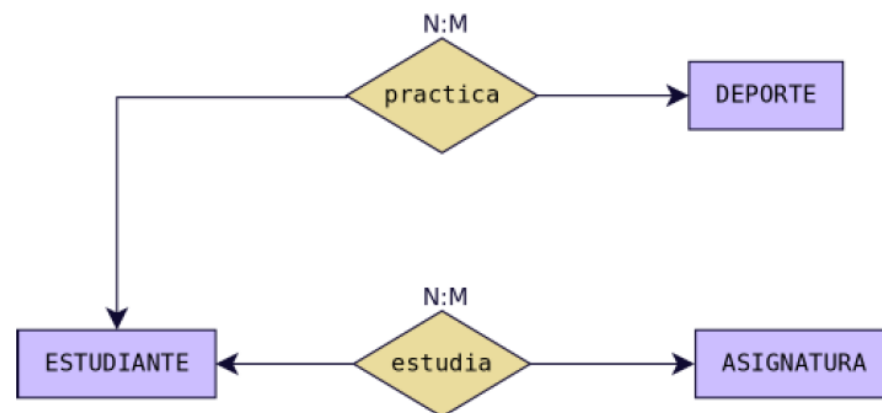
Por otro lado no existe ninguna dependencia entre la asignatura cursada y el deporte practicado.

Para normalizar a 4FN creamos 2 tablas:

Estudiante	Asignatura
11111111A	Matemáticas
11111111A	Lengua
22222222B	Matemáticas

Estudiante	Deporte
11111111A	Natación
11111111A	Baloncesto
22222222B	Fútbol
22222222B	Natación

Diagrama E/R equivalente



VI. QUINTA FORMA NORMAL: 5FN

La quinta forma normal (5FN), es una generalización de la anterior. También conocida como forma normal de proyección-uni3n (PJ/NF). Una tabla se dice que est1 en 5NF si y s3lo si est1 en 4NF y cada dependencia de uni3n (join) en ella es implicada por las claves candidatas.

Ejemplo: Tenemos una tabla con varios proveedores que nos proporcionan piezas para distintos proyectos. Asumimos que un Proveedor suministra ciertas Piezas en particular, un Proyecto usa ciertas Piezas, y un Proyecto es suplido por ciertos Proveedores, entonces tenemos las siguientes dependencias multivaluadas:

- Proveedor \twoheadrightarrow Pieza
- Pieza \twoheadrightarrow Proyecto
- Proyecto \twoheadrightarrow Proveedor

Se puede observar como se produce un ciclo:

- Proveedor \twoheadrightarrow Pieza \twoheadrightarrow Proyecto \twoheadrightarrow Proveedor (nuevamente)

VI. QUINTA FORMA NORMAL: 5FN

Suministros

Proveedor	Pieza	Proyecto
E1, E4, E6	PI3, PI6	PR2, PR4
E2, E5	PI1, PI2	PR1, PR3
E3, E7	PI4, PI5	PR5, PR6

Suministros

Proveedor	Pieza	Proyecto
E1	PI3	PR2
E1	PI3	PR4
E1	PI6	PR2
E1	PI6	PR4
E4	PI3	PR2
E4	PI3	PR4
E4	PI6	PR2
E4	PI6	PR4
E6	PI3	PR2
E6	PI3	PR4
E6	PI6	PR2
E6	PI6	PR4
E2	PI1	PR1
E2	PI1	PR3
E2	PI2	PR1
E2	PI2	PR3
E5	PI1	PR1
E5	PI1	PR3
E5	PI2	PR1
E5	PI2	PR3
E3	PI4	PR5
E3	PI4	PR6
E3	PI5	PR5
E3	PI5	PR6
E7	PI4	PR5
E7	PI4	PR6
E7	PI5	PR5
E7	PI5	PR6

Descomponemos la tabla en 3 tabla nuevas:

- Proveedor-Pieza,
- Pieza-Proyecto
- Proyecto-Proveedor

Proveedor-Pieza

Proveedor	Pieza
E1	PI3
E1	PI6
E4	PI3
E4	PI6
E6	PI3
E6	PI6
E2	PI1
E2	PI2
E5	PI1
E5	PI2
E3	PI4
E3	PI5
E7	PI4
E7	PI5

Pieza-Proyecto

Pieza	Proyecto
PI3	PR2
PI3	PR4
PI6	PR2
PI6	PR4
PI1	PR1
PI1	PR3
PI2	PR1
PI2	PR3
PI4	PR5
PI4	PR6
PI5	PR5
PI5	PR6

Proyecto-Proveedor

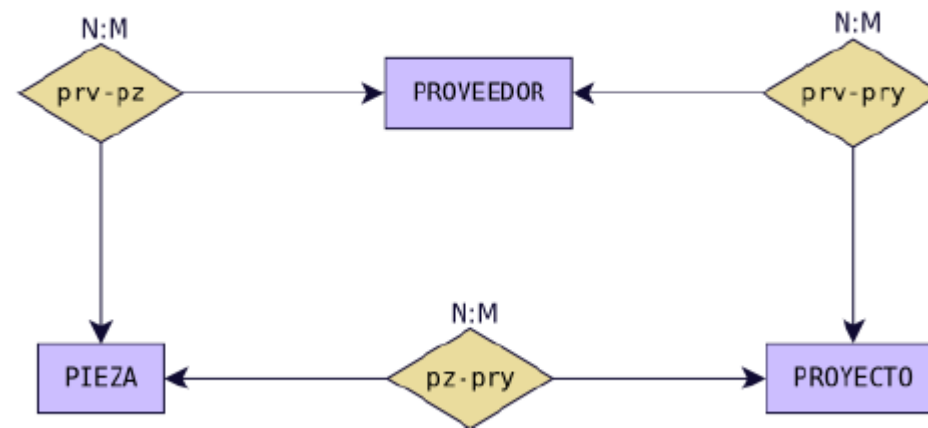
Proyecto	Proveedor
PR2	E1
PR4	E1
PR2	E4
PR4	E4
PR2	E6
PR4	E6
PR1	E2
PR3	E2
PR1	E5
PR3	E5
PR5	E3
PR6	E3
PR5	E7
PR6	E7

VI. QUINTA FORMA NORMAL: 5FN

El producto natural de estas 3 tablas nos da la tabla original.

Proveedor-Pieza |x| Pieza-Proyecto |x| Proyecto-Proveedor = Suministros

Diagrama E/R equivalente :



VII. DESCOMPOSICIÓN DE RELACIONES

La transformación de una relación que se encuentra en una determinada forma normal en otra relación cuya forma normal es superior se realiza por medio del **operador *proyección*** del álgebra relacional.

Así, por ejemplo, la relación:

RELACIÓN (Campo1, Campo2, Campo3)

es tal que se encuentra en 1FN por que su único atributo no principal (Campo3) no depende totalmente de la clave (agregación de Campo1 y de Campo2), sino de parte de ella (por ejemplo, Campo3 solamente depende de Campo2), puede llevarse a una forma normal más avanzada descomponiéndola mediante proyecciones, obteniendo así varias relaciones:

$$\begin{aligned} RELACIÓN1 &= \pi_{Campo1, Campo2}(RELACIÓN) \\ RELACIÓN2 &= \pi_{Campo2, Campo3}(RELACIÓN) \end{aligned}$$

VII. DESCOMPOSICIÓN DE RELACIONES

Estando ambas relaciones en una forma normal superior. En concreto en 3FN ya que la combinación natural $RELACIÓN1 * RELACIÓN2$ mediante el atributo común Campo2 devuelve la relación original $RELACIÓN$

Ejemplo completo:

Normalizar la siguiente relación o tabla:

PIEZAS(cod_pieza, nombre_pieza, precio_pieza, cod_almacen, nombre_almacén, dir_almacén, encargado_almacén, fax_encargado, email_encargado)

Teniendo en cuenta:

- Una pieza se puede encontrar en diferentes almacenes.
- Un almacén sólo tiene un encargado
- Una persona se puede encargar de varios almacenes

VII. DESCOMPOSICIÓN DE RELACIONES

¿Está en 1FN? Sí porque todos los campos son atómicos

PIEZAS (cod_pieza, nombre_pieza, precio_pieza, cod_almacén, nombre_almacén, dir_almacén, encargado_almacén, fax_encargado, email_encargado) 1FN

¿está en 2FN? **No**

- porque nombre_almacén, dir_almacén, encargado_almacén, fax_encargado, email_encargado no dependen completamente de la clave principal (cod_pieza, cod_almacén sino de parte de ella cod_almacén)

- porque nombre_pieza, precio_pieza no dependen completamente de la clave principal (cod_pieza, cod_almacén sino de parte de ella cod_pieza)

VII. DESCOMPOSICIÓN DE RELACIONES

Cod_pieza, cod_almacén → nombre_pieza, precio_pieza, nombre_almacén, dir_almacén, encargado_almacén, fax_encargado, email_encargado) **no DF completa**

Entonces:

cod_almacén → nombre_almacén, dir_almacén, encargado_almacén, fax_encargado, email_encargado

cod_pieza → nombre_pieza, precio_pieza

PIEZAS-ALMACÉN (cod_pieza, cod_almacén) 2FN

PIEZAS (cod_pieza, nombre_pieza, precio_pieza) 2FN

ALMACÉN (cod_almacén, nombre_almacén, dir_almacén, encargado_almacén, fax_encargado, email_encargado) 2FN

VII. DESCOMPOSICIÓN DE RELACIONES

¿está **PIEZAS-ALMACÉN** en 3FN?

Sí porque no existen atributos que no pertenezcan a la clave principal

¿está **ALMACÉN** en 3FN?

No porque

$\text{cod_almacén} \rightarrow \text{encargado_almacén} \rightarrow \text{fax_encargado}, \text{email_encargado}$

$\text{encargado_almacén} \twoheadrightarrow \text{cod_almacén}$

ALMACÉN (cod_almacén, nombre_almacén, dir_almacén, **encargado_almacén**)

ENCARGADOS (nom_encargado, fax_encargado, email_encargado)

VII. DESCOMPOSICIÓN DE RELACIONES

¿está PIEZAS en 3FN?

Sí porque ningún atributo que no forma parte de la clave principal depende transitivamente de ésta.

ESQUEMA RELACIONAL NORMALIZADO

PIEZAS-ALMACÉN (cod_pieza, cod_almacén) 3FN

PIEZAS (cod_pieza, nombre_pieza, precio_pieza) 3FN

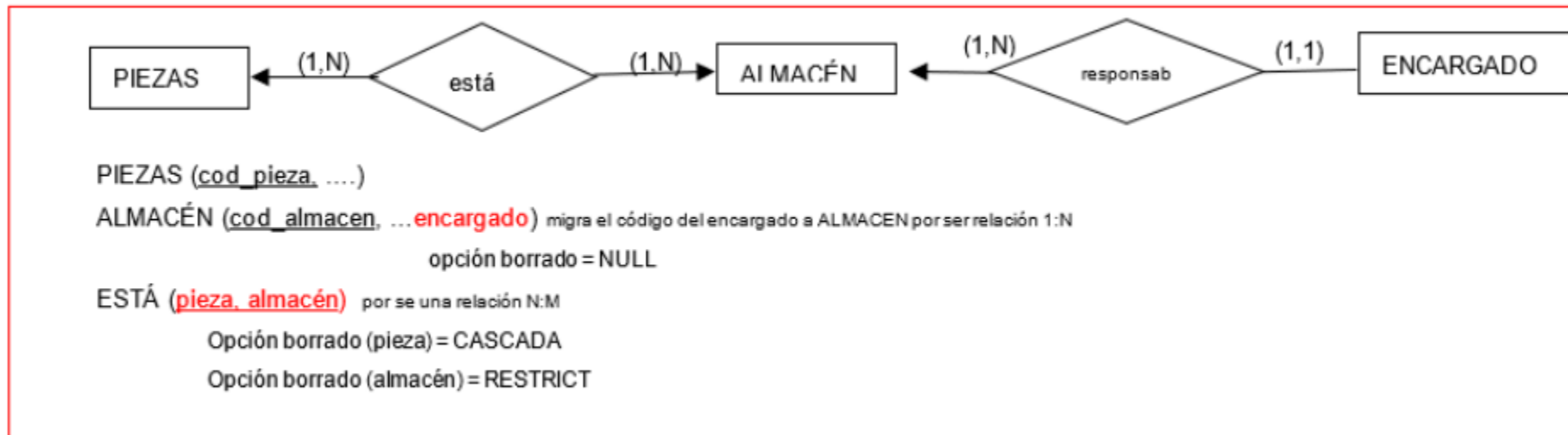
ALMACÉN (cod_almacén, nombre_almacén, dir_almacén, encargado_almacén) 3FN

ENCARGADOS (nom_encargado, fax_encargado, email_encargado) 3FN

VII. DESCOMPOSICIÓN DE RELACIONES

DESCOMPOSICIÓN SIN PÉRDIDA DE DEPENDENCIAS FUNCIONALES

Las dependencias funcionales recogen la semántica del mundo real por lo que es conveniente conservarlas en el proceso de descomposición



VII. DESCOMPOSICIÓN DE RELACIONES

DESCOMPOSICIÓN EN PROYECCIONES INDEPENDIENTES

La descomposición de una relación R en un conjunto de relaciones $\{R_i\}$ se dice que se ha realizado en proyecciones independientes si **no ha habido pérdida de información ni pérdida de dependencias funcionales**.

Se trata de la mejor descomposición ya que las relaciones resultantes son equivalentes a la relación original y, en ellas se han eliminado las anomalías de inserción, modificación y borrado.

Toda relación en 3FN puede descomponerse sin pérdida de información ni de dependencias funcionales.

VIII. ÁLGEBRA RELACIONAL

El modelo relacional es la forma de representar los datos y manipular dicha representación considerando la integridad de los datos.

Estos, en el modelo relacional se representan mediante Relaciones y un procedimiento para su manipulación es el **Álgebra Relacional**.

El *Álgebra relacional* es un lenguaje de consulta procedimental.

Un *lenguaje de consulta* es aquel que el usuario utiliza para solicitar información de la base de datos. Este tipo de lenguajes puede ser:

- *Procedimental*.

El usuario indica la secuencia de operaciones que debe realizar sobre la base de datos para obtener el resultado deseado

- *No procedimental*.

El usuario describe la información que desea sin dar el procedimiento para determinarla.

VIII. ÁLGEBRA RELACIONAL

Para realizar las consultas a la base de datos, el álgebra relacional dispone de:

- La *operación de asignación* asigna el valor de alguna expresión del álgebra a una relación nombrada.
- Los *operadores* toman una o dos relaciones como entrada y producen una nueva relación como salida. En álgebra relacional se definen ocho tipos de operadores básicos:
 - Operaciones de Conjuntos: Unión, Intersección, Diferencia y Producto Cartesiano.
 - Operaciones relacionales: Selección, Proyección, Reunión y División.

Además de estos operadores, se define el operador Renombrar que permite cambiar el nombre de los atributos de una relación.

VIII. ÁLGEBRA RELACIONAL

Operador Renombrar (ρ)

Cambia el nombre de los atributos que sean necesarios en una relación antes de realizar una operación que pueda llevar a una relación con una cabecera en la que aparezcan dos atributos con el mismo nombre.

A partir de una relación especificada crea una nueva copia de ésta en la que sólo se han modificado los nombres de aquellos atributos que se quieren renombrar.

La sintaxis, para una relación de nombre R, es:

$R \rho_{\text{Atributos Nuevos}} (\text{Atributos originales})$

VIII. ÁLGEBRA RELACIONAL

Unión (U)

La unión de dos relaciones R1 y R2 compatibles respecto a la unión es una nueva relación R cuya cabecera es idéntica a la de las dos relaciones y cuyo cuerpo está formado por todas las tuplas pertenecientes a R1, a R2 o a las dos. Esto es, está formado por todas las tuplas que aparecen en cualquiera de las dos relaciones especificadas. Su sintaxis es:

$$R = R1 \cup R2$$

$$R = R_1 \cup R_2$$

Tabla A

NºMat	Nombre
111	Alfredo
112	Carolina
114	José

Tabla B

NºMat	Nombre
112	Carolina
113	Nuria
115	Antonio



Aplicando
la Unión

Tabla A U B

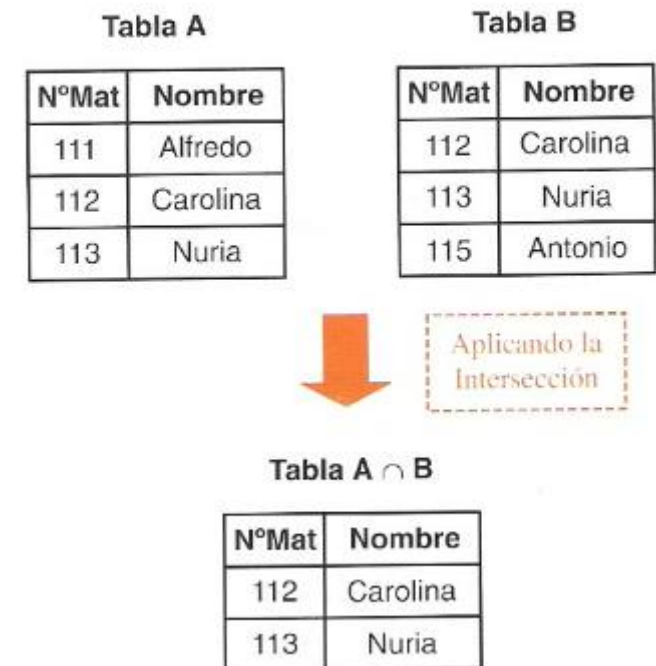
NºMat	Nombre
111	Alfredo
112	Carolina
113	Nuria
114	José
115	Antonio

VIII. ÁLGEBRA RELACIONAL

Intersección (\cap)

La intersección de dos relaciones R1 y R2 compatibles respecto a la unión es una nueva relación R cuya cabecera es idéntica a la de las dos relaciones y cuyo cuerpo está formado por todas las tuplas pertenecientes tanto a R1, como a R2 o a las dos. Esto es, está formado por todas las tuplas que aparecen en las dos relaciones especificadas. Su sintaxis es:

$$R = R1 \cap R2$$

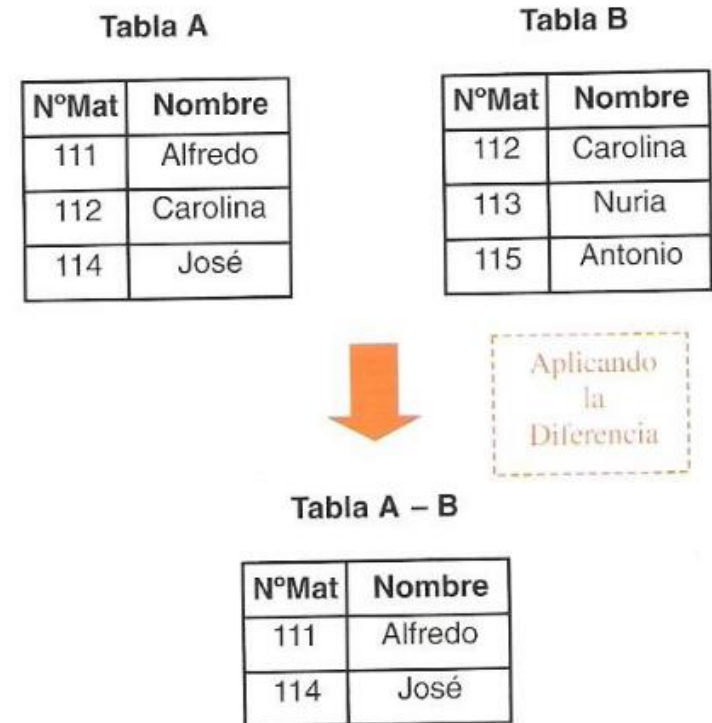


VIII. ÁLGEBRA RELACIONAL

Diferencia (-)

La diferencia de dos relaciones R1 y R2 compatibles respecto a la unión es una nueva relación R cuya cabecera es idéntica a la de las dos relaciones y cuyo cuerpo está formado por todas las tuplas pertenecientes a R1 pero no a R2. Esto es, está formado por todas las tuplas de la primera relación que no aparecen en la segunda. Su sintaxis es:

$$R = R1 - R2$$

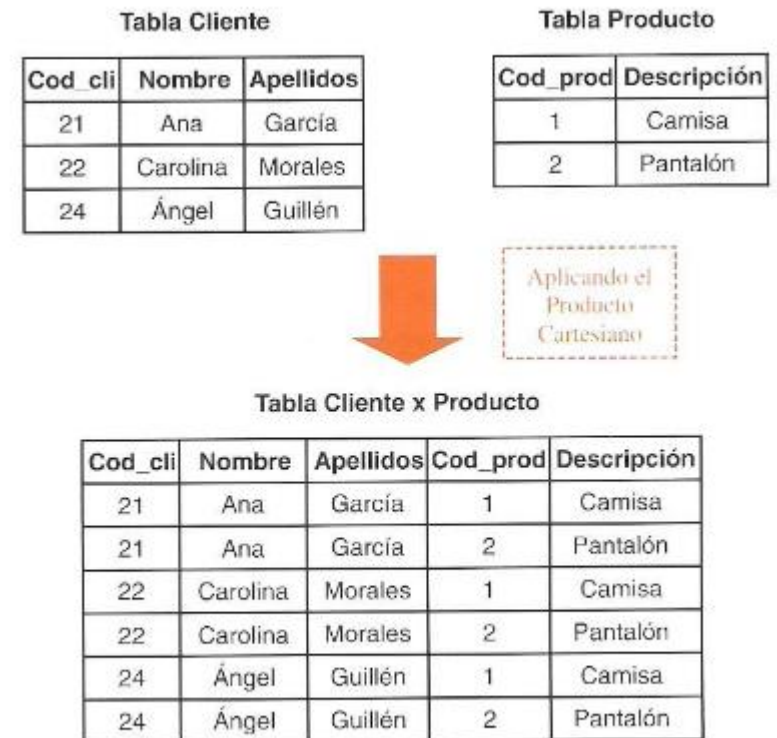


VIII. ÁLGEBRA RELACIONAL

Producto Cartesiano (x)

El producto cartesiano de dos relaciones R1 y R2 compatibles respecto al producto es una nueva relación R cuya cabecera es una combinación de las cabeceras de R1 y R2 y cuyo cuerpo está formado por el conjunto de todas las tuplas t tales que t es la combinación de la tupla t1 perteneciente a R1 y la tupla t2 perteneciente a R2. Esto es, está formado por todas las combinaciones posibles de tuplas, una de cada una de las dos relaciones. Su sintaxis es:

$$R = R1 \times R2$$



VIII. ÁLGEBRA RELACIONAL

Producto Cartesiano (x)

El producto cartesiano de dos relaciones R1 y R2 compatibles respecto al producto es una nueva relación R cuya cabecera es una combinación de las cabeceras de R1 y R2 y cuyo cuerpo está formado por el conjunto de todas las tuplas t tales que t es la combinación de la tupla t1 perteneciente a R1 y la tupla t2 perteneciente a R2. Esto es, está formado por todas las combinaciones posibles de tuplas, una de cada una de las dos relaciones. Su sintaxis es:

$$R = R1 \times R2$$

