

Unidad 4

# DISEÑAR UNA ESTRUCTURA DE DATOS EFICIENTE

Prof. Mg. Ing. Roxana Martínez  
Prof. Esp. Lic. Silvia Salgado



**UAIOnline**  
**Ultra**»»



# Diseño de una estructura de datos eficiente

## Unidad 4

### ■ OBJETIVOS

Dependencias funcionales y normalización por Bases de Datos Relacionales. Pautas informales de diseño. Primera, segunda y tercera forma normales. Proceso para la normalización. Determinantes. Forma Normal de Boyce / Codd.



# INTRODUCCIÓN

Cuando se diseña una base de datos se busca generar un conjunto de esquemas que permitan almacenar información ***sin redundancias***, permitiendo modificaciones sin ambigüedades y, por supuesto, facilitando el recupero de dicha información en ***forma eficiente***.

# INTRODUCCIÓN

En general, se supone que partiendo de un buen trabajo de relevamiento y de un buen diagrama de entidad-relación debería llegarse en forma directa a un diseño correcto de una base de datos.

Sin embargo, no siempre nos vamos a encontrar involucrados en la etapa de diseño. Es probable que nos presenten un sistema ya armado para corregir o bien que nos pidan migrar un sistema hacia otra plataforma, siendo responsables de efectuarle los cambios pertinentes para que luego funcione correctamente.

# ANOMALÍAS DE ACTUALIZACIÓN

***Base de datos relacional*** es un conjunto de esquemas relacionados y sus restricciones.

La ***mala representación de la información***, como puede ser la redundancia de datos, produce ciertas anomalías en la actualización (inserción, modificación y borrado) de los mismos.

## EJEMPLO:

Base mal diseñada y evidenciaremos las distintas anomalías

ALUMNO					
legajo	nombre	direccion	nroActa	nota	Materia
400	'Ana Lopez'	Santa Fé 235	1103	2	Programacion I
400	'Ana Lopez'	Santa Fé 235	1104	2	Programacion I
400	'Ana Lopez'	Santa Fé 235	1108	7	Algebra Lineal
1300	'Raul Sosa'	Gurruchaga 3015	1333	9	Matematica II
1300	'Raul Sosa'	Gurruchaga 3015	1103	2	Programacion I
1300	'Raul Sosa'	Gurruchaga 3015	1106	4	Programacion I
1300	'Raul Sosa'	Gurruchaga 3015	1108	5	Algebra Lineal
2000	'Ana Lopez'	Maure 1925	1103	2	Programacion I

## EJEMPLO:

Supongamos que el alumno Raúl Sosa rinde un examen de Programación II y resulta ser evaluado con 7 puntos, quedando constancia en el acta 1406, para lo cual deberíamos insertar la tupla:

***( 1300, 'Raul Sosa', Gurruchaga 3015, 1406, 7, Programación II )***

Es evidente que estamos repitiendo información, ya que el nombre y la dirección del alumno con legajo 1300 resultan ajenos al examen que haya rendido el mismo. Dicha repetición, además de malgastar espacio, complica la inserción. Se trata de una redundancia de datos.

## TIPOS DE ANOMALÍAS

Otro problema de nuestro diseño es que no se puede tener información de un alumno que no haya rendido algún examen, a menos que se completen número de Acta, nota y materia con NULL o datos ficticios.

El inconveniente radica en que la información inherente al alumno debería estar disponible independientemente de que éste haya rendido o no exámenes. Estamos frente a la llamada ***anomalía de inserción***.



# TIPOS DE ANOMALÍAS

Hay otro ***problema grave que está ligado a la modificación de la información.***

Pensemos que el alumno Sosa cambia su dirección. Con este diseño estaríamos obligados a realizar el cambio en todas las tuplas donde figure Raúl Sosa, complicándose la actualización de la información y corriendo el riesgo de incurrir en alguna inconsistencia. Este hecho evidencia la llamada ***anomalía de modificación.***

Por otra parte, pensemos que si se borrarán todas las apariciones de Programación I, perderíamos simultáneamente toda la información respectiva a la alumna Ana López, de legajo 2000. Esto presenta una muestra de la llamada ***anomalía de borrado.***

# DISEÑO DE BASE DE DATOS

Intuitivamente, lo que ocurre es que para un determinado legajo sólo hay un nombre y una dirección posibles, es decir, nombre y dirección dependen de legajo.

En cambio, un mismo alumno (identificado por legajo) puede haber dado muchos exámenes, con lo cual un mismo legajo puede tener muchos números de acta asociados. Evidentemente se trata de un mal diseño.

# DISEÑO DE BASE DE DATOS

El objetivo de un buen diseño de una base de datos es ***eliminar toda redundancia de datos y reducir o eliminar anomalías de manipulación de datos.***

La ***Teoría de Normalización*** es una teoría completa que le permite al diseñador eliminar problemas de diseño por medio de la descomposición o reorganización de la estructura de tablas existentes.

# DEPENDENCIAS FUNCIONALES:

## Dependencia Funcional :

Sean  $X$  e  $Y$  dos conjuntos de atributos de un esquema de relación  $R$ .

Se dice que  $X$  determina funcionalmente a  $Y$  o que  $Y$  depende funcionalmente de  $X$ , lo cual se denota  $X \rightarrow Y$ , si se cumple:

Que para dos tuplas  $t_1, t_2$  en una relación  $r$ ,

si se cumple que  $t_1[X] = t_2[X]$  entonces  $t_1[Y] = t_2[Y]$

## DEPENDENCIAS FUNCIONALES - EJEMPLOS:

***En un sistema de biblioteca: ISBN → tituloLibro*** (no puede haber dos libros de igual ISBN con distinto título).

***En un sistema estatal: nroCuil → nombre*** (dos personas distintas no pueden tener el mismo CUIL).

***En un sistema de facturación: articulo cantidad → montoVenta***

La venta de la misma cantidad de un determinado artículo no puede acusar distintos montos (sin tener en cuenta descuentos).

***Las dependencias funcionales son una propiedad de la semántica de los atributos, ya que tienen relación con el significado de dichos datos y de cómo se influyen entre sí.***

# INDEPENDENCIA DE LOS DATOS

Es el objetivo esencial de un DBMS.

Consiste en mantener la inmunidad de las aplicaciones ante los cambios en la estrategia de acceso o en la estructura de almacenamiento de los datos.



# DEPENDENCIAS FUNCIONALES

## Ejemplo:

Para el esquema dado como ejemplo, podemos determinar el conjunto de dependencias funcionales  $F$  a través del significado de los atributos.

$F = \{ \text{legajo} \rightarrow \text{nombre}, \text{legajo} \rightarrow \text{dirección}, \text{nroActa} \rightarrow \text{materia}, \text{legajo} + \text{nroActa} \rightarrow \text{nota} \}$

# DEPENDENCIAS FUNCIONALES

También podemos determinar que tiene clave compuesta: ***legajo + nroActa***

Sin embargo hay muchas otras dependencias funcionales que también se cumplen, como por ejemplo:

***legajo → nombre + dirección***

***legajo + nroActa → materia***

***legajo + nombre → nombre***

***etc.***



# FORMAS NORMALES

## TABLAS

**1º FN**

**2º FN**

**3º FN**

**FN BOYCE/CODD**

**4º FN**

**5º FN o FN PR**

# PRIMERA FORMA NORMAL ( FN1)

- Una relación está en FN1 si y solo si todos sus dominios subyacentes sólo contienen valores atómicos.
- Una relación está en FN1 si no tiene grupos repetitivos

# PRIMERA FORMA NORMAL ( FNI)

Columnas cuyos Datos --  
presentan sólo un valor

Columnas que presentan Datos --  
con más de un valor !! No atomicos!

IdPedido	Fecha	IdCliente	NombreCli	IdProducto	NombrePnd	Cantidad
123	23/11/1998	010	E.Metálicas	P2 P4 P9	Lic.Contab Lic. Planilla y RRHH Mantenimiento Anual P2	20 5 1
246	13/10/1998	020	M.Soldadura	P2 P9	Lic.Contab Mantenimiento Anual P2	10 1
280	5/12/1998	010	E.Metálicas	P8 P12	Mantenimiento Anual P4 Lotus Notes	1 5

# PRIMERA FORMA NORMAL ( FNI )

Para pasar a FNI  
debo cortar por alli,

<u>IdPedido</u>	Fecha	IdCliente	NombreCli	IdProducto	NombrePrd	Cantidad
123	23/11/1998	010	E.Metálicas	P2	Lic.Contab	20
				P4	Lic. Planilla y RRHH	5
				P9	Mantenimiento Anual P2	1
246	13/10/1998	020	M.Soldadura	P2	Lic.Contab	10
				P9	Mantenimiento Anual P2	1
280	5/12/1998	010	E.Metálicas	P8	Mantenimiento Anual P4	1
				P12	Lotus Notes	5



Grupo repetitivo

# PRIMERA FORMA NORMAL ( FNI )

Al pasarlo a FNI de esa tabla original desnormalizada obtengo el siguiente par de tablas normalizadas:

Las tablas obtenidas deben relacionarse para que yo pueda volver a obtener la informacion original

<u>IdPedido</u>	<u>Fecha</u>	<u>IdCliente</u>	<u>NombreCli</u>
123	23/11/1998	010	E.Metálicas
246	13/10/1998	020	M.Soldadura
280	5/12/1998	010	E.Metálicas

Pedido

necesitó clave anterior

<u>IdPedido</u>	<u>IdProducto</u>	<u>NombrePrd</u>	<u>Cantidad</u>
123	P2	Lic.Contab	20
123	P4	Lic. Planilla y RRHH	5
123	P9	Mantenimiento Anual P2	1
246	P2	Lic.Contab	10
246	P9	Mantenimiento Anual P2	1
280	P8	Mantenimiento Anual P4	1
280	P12	Lotus Notes	5

Detalle de pedido

## SEGUNDA FORMA NORMAL (FN2)

A partir de aquí retomamos con los conceptos de dependencias funcionales vistos anteriormente:

***Una relación está en FN2 si y solo si está en FN1 y cada atributo no primo (o sea que no conforma la clave primaria) es completamente dependiente de la clave primaria.***

Es decir:

Una relación está en FN2 si está en FN1 y todos los atributos no primos dependen de la clave completa.

## SEGUNDA FORMA NORMAL (FN2)

**NombrePrd** depende de **Idproducto**  
(sólo una parte de la clave)

clave

<u>IdPedido</u>	<u>IdProducto</u>	<u>NombrePrd</u>	Cantidad
123	P2	Lic. Contab	20
123	P4	Lic. Planilla y RRHH	5
123	P9	Mantenimiento Anual P2	1
246	P2	Lic. Contab	10
246	P9	Mantenimiento Anual P2	1
280	P8	Mantenimiento Anual P4	1
280	P12	Lotus Notes	5

Detalle de pedido

## SEGUNDA FORMA NORMAL (FN2)

*Una vez realizado el pasaje a FN2 obtenemos de la tabla desnormalizada original este par de tablas normalizadas:*

**Notar nuevamente que:**

La parte de la clave IdProducto de la cual dependía el atributo nombreProd, ahora se convierte en clave de una nueva tabla.

Las nuevas tablas se relacionaran a través del atributo idProducto.

Detalle de pedido

IdPedido	IdProducto	Cantidad
123	P2	20
123	P4	5
123	P9	1
246	P2	10
246	P9	1
280	P8	1
280	P12	5

Producto

IdProducto	NombreProd
P2	Lic.Contab
P4	Lic. Planilla y RRHH
P9	Mantenimiento Anual P2
P8	Mantenimiento Anual P4
P12	Lotus Notes



## TERCERA FORMA NORMAL (FN3)

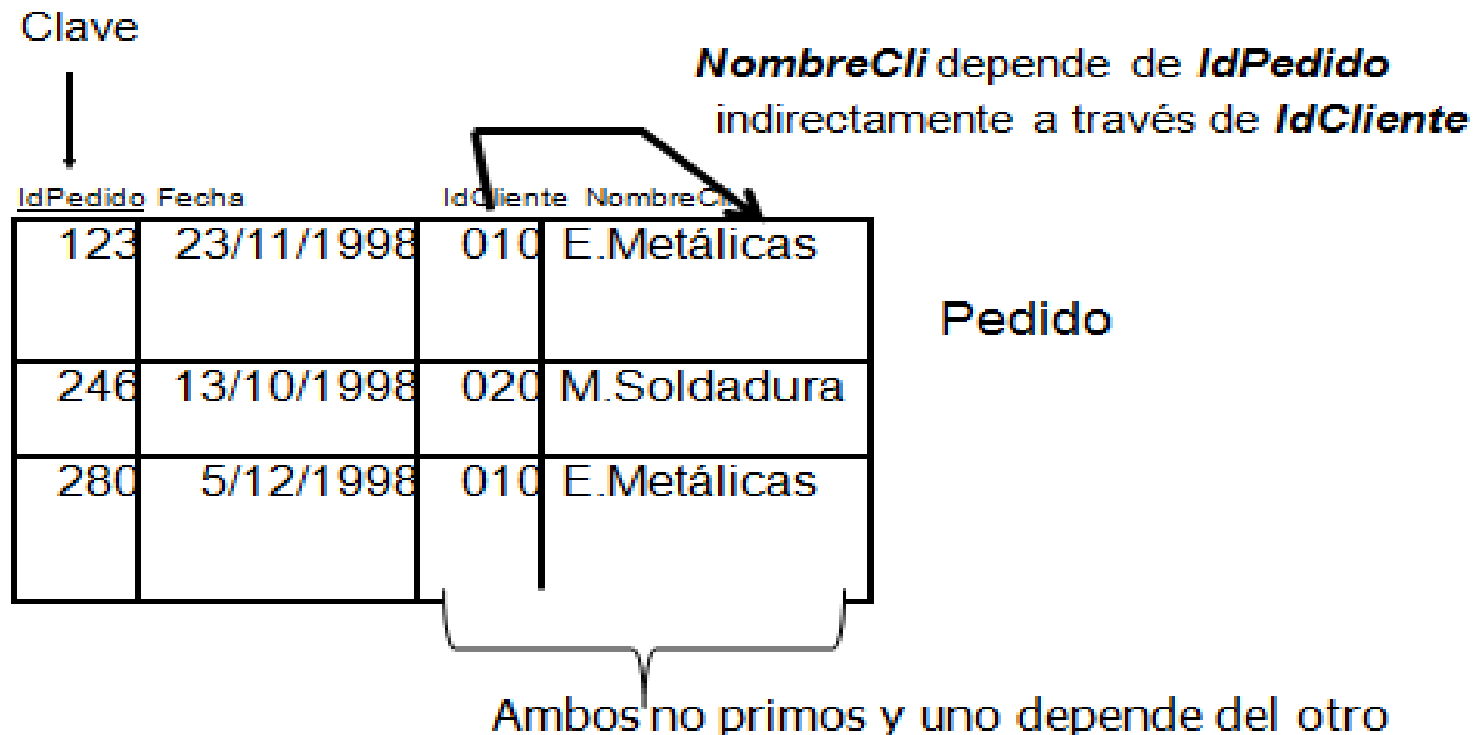
- Una relación está en FN3 si y solo si está en FN2 y ningún atributo no primo (los que no forman parte de la clave primaria) depende de otro atributo no primo.
- Una relación está en FN3 si y solo si está en FN2 y ningún atributo depende de la clave a través de otro atributo no primo (transitividad)
- Dependencia transitiva:

$$X \rightarrow Y \text{ solo si } X \rightarrow Z \text{ y } Z \rightarrow Y$$

# TERCERA FORMA NORMAL (FN3)

Dependencia transitiva:

$$X \rightarrow Y \text{ solo si } X \rightarrow Z \text{ y } Z \rightarrow Y$$



X es idPedido y determina a Y que es idCliente. Pero también X determina a Z que es NombreCliente.

Por lo tanto vemos que también se cumple que Y determina a Z. Esta es la transitividad en esta tabla desnormalizada.

## TERCERA FORMA NORMAL (FN3)

Una vez realizado el pasaje a FN3 obtendremos las siguientes tablas:

Notar que el atributo no primo idCliente de la tabla original desnormalizada, pasa a ser atributo clave en una nueva tabla y ambas tablas se relacionaran a través del mismo.

Pedido

<u>IdPedido</u>	Fecha	IdCliente
123	23/11/1998	010
246	13/10/1998	020
280	5/12/1998	010

Cliente

<u>IdCliente</u>	NombreCli
010	E.Metálicas
020	M.Soldadura

# ESQUEMA FINAL NORMALIZADO DEL EJEMPLO

<u>IdPedido</u>	Fecha	IdCliente
123	23/11/1998	010
246	13/10/1998	020
280	5/12/1998	010

Pedido

<u>IdCliente</u>	NombreCli
010	E.Metálicas
020	M.Soldadura

Cliente

Detalle de pedido

<u>IdPedido</u>	<u>IdProducto</u>	Cantidad
123	P2	20
123	P4	5
123	P9	1
246	P2	10
246	P9	1
280	P8	1
280	P12	5

Producto

<u>IdProducto</u>	NombrePrd
P2	Lic.Contab
P4	Lic. Planilla y RRHH
P9	Mantenimiento Anual P2
P8	Mantenimiento Anual P4
P12	Lotus Notes

## FORMA NORMAL DE BOYCE CODD (FNBC)

Es más estricta que la FN3. Si una relación está en FNBC entonces esta en FN3, pero que esté en FN3 no asegura que esté en FNBC.

En términos menos formales, una tabla está en FNBC si está en 3FN y los únicos determinantes son las claves candidatas.

# FORMA NORMAL DE BOYCE CODD

Supongamos la siguiente relación:

Titulo	Año	Longitud	Estudio	Tipo
Star wars	1977	124	Fox	color
Mighty Ducks	1991	104	Disney	color
El mundo de Wayne	1992	95	Paramount	color

## FORMA NORMAL DE BOYCE CODD

**No está en FNBC porque...**

Primero veremos cuales son claves aquí.

La clave es Titulo+año

**Por lo tanto cualquier conjunto de atributos que la contengan es una superclave.**

**Ejemplos de superclaves:**

Titulo+año+longitud

Titulo+año+tipo

Titulo+año+tipo+longitud

Etc...

# FORMA NORMAL DE BOYCE CODD

## *Dependencias Funcionales:*

Titulo + año determinan longitud, tipo y estudio

**Pero tanto longitud, como tipo como estudio determinan titulo y también año**

Por lo tanto no esta en FNBC, porque estos atributos son no primos y están determinando a atributos que no pueden descomponerse en otras tablas dependiendo de ellos (por formar parte de la clave candidata no podemos quitarlos de la tabla).





¿Consultas?



**UAI**

**Universidad Abierta  
Interamericana**