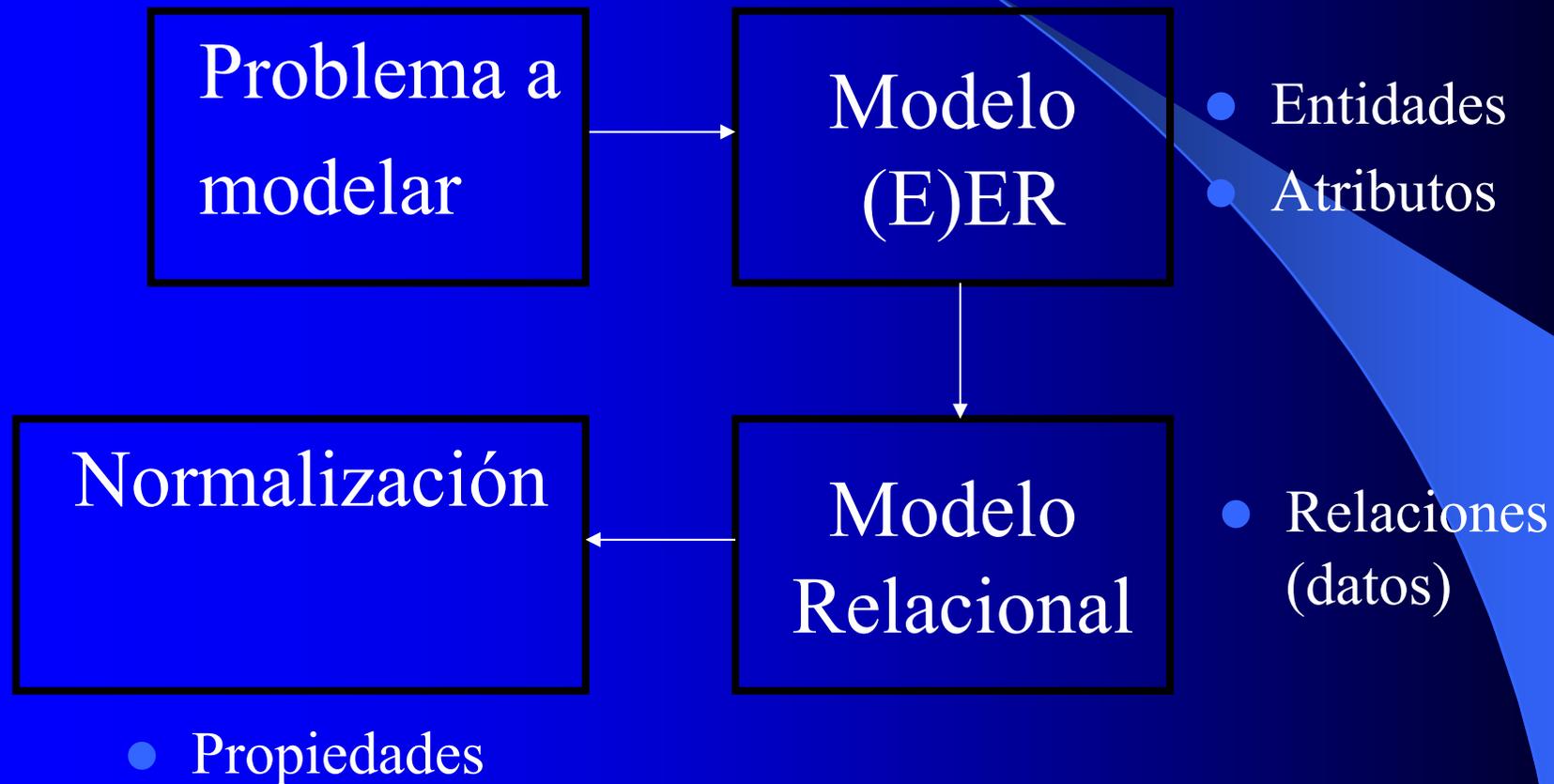


# “Práctica” de la Normalización

Sergio Ilarri

# Contexto (I)



# Contexto (II)



# Diseño de BDs

- Dos aproximaciones:
  - Bottom-up (síntesis)
  - Top-down (análisis)



# Relaciones

- DiseñaBD:
  - Entrada: conjunto de atributos
  - Salida:
    - conjunto de relaciones
    - atributos de cada relación
- ¿Todas las tablas son relaciones?:
  - Nombre único
  - Atributos monovaluados
  - Filas únicas
  - Atributos (columnas) con nombre único
  - Orden de filas/columnas irrelevantes

Requisitos  
(genéricos)

# Joins de Relaciones

Customer ID	Company Name	Contact	Phone Number	Credit Limit
-------------	--------------	---------	--------------	--------------

Invoice ID	Invoice Date	Order Date	Customer ID	Employee ID	Customer PO
------------	--------------	------------	-------------	-------------	-------------

Invoice ID	Inventory ID	Quantity	Unit Price	Discount
------------	--------------	----------	------------	----------

Inventory ID	Item ID	Caffeinated	Price	On Hand
--------------	---------	-------------	-------	---------

# Parece Fácil, Pero...

- Problemas:

- Redundancia

- Anomalías:

- Actualización

- Inserción

- Borrado



- Creación

- Mantenimiento

- Modificación

- Solución:

- Normalizar (identificar y eliminar anomalías)

# Objetivo

- Mejorar y validar el diseño lógico
- Evitar duplicaciones de datos:
  - descomposición de relaciones
- Desarrollado inicialmente por E.F. Codd
- Relaciones bien estructuradas
- Normalización: proceso consistente en asegurar que cada tabla trata de un solo concepto

# ¿Es Necesaria?

- Un modelo E/R bien diseñado evita la necesidad de usarla

**PERO...**

- Guía para evitar fallos (principiantes)
- Modo de probar la corrección del diseño
- Formalizan el sentido común
- Posibilidad de automatización

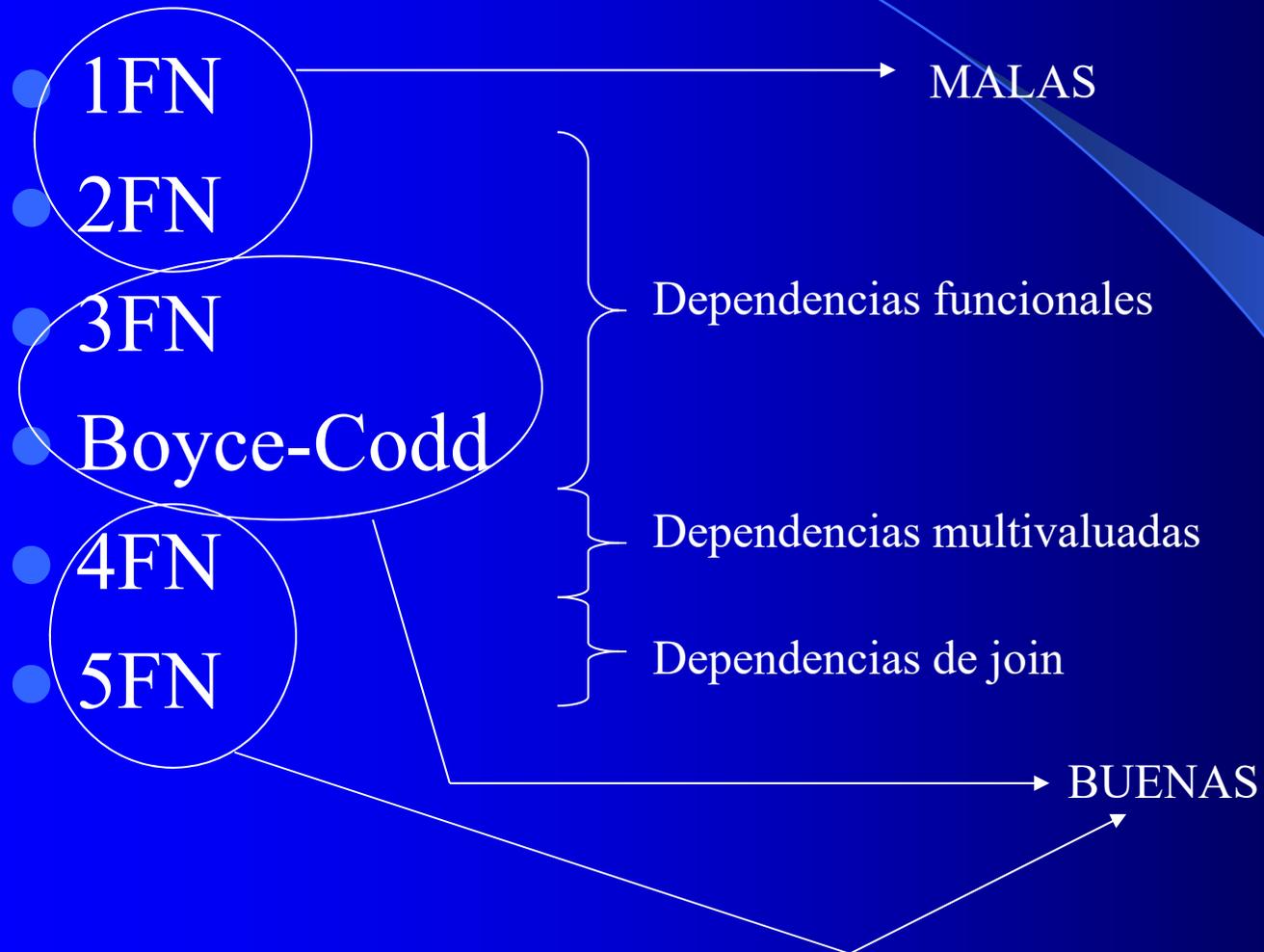
# Ejemplo

## Employees

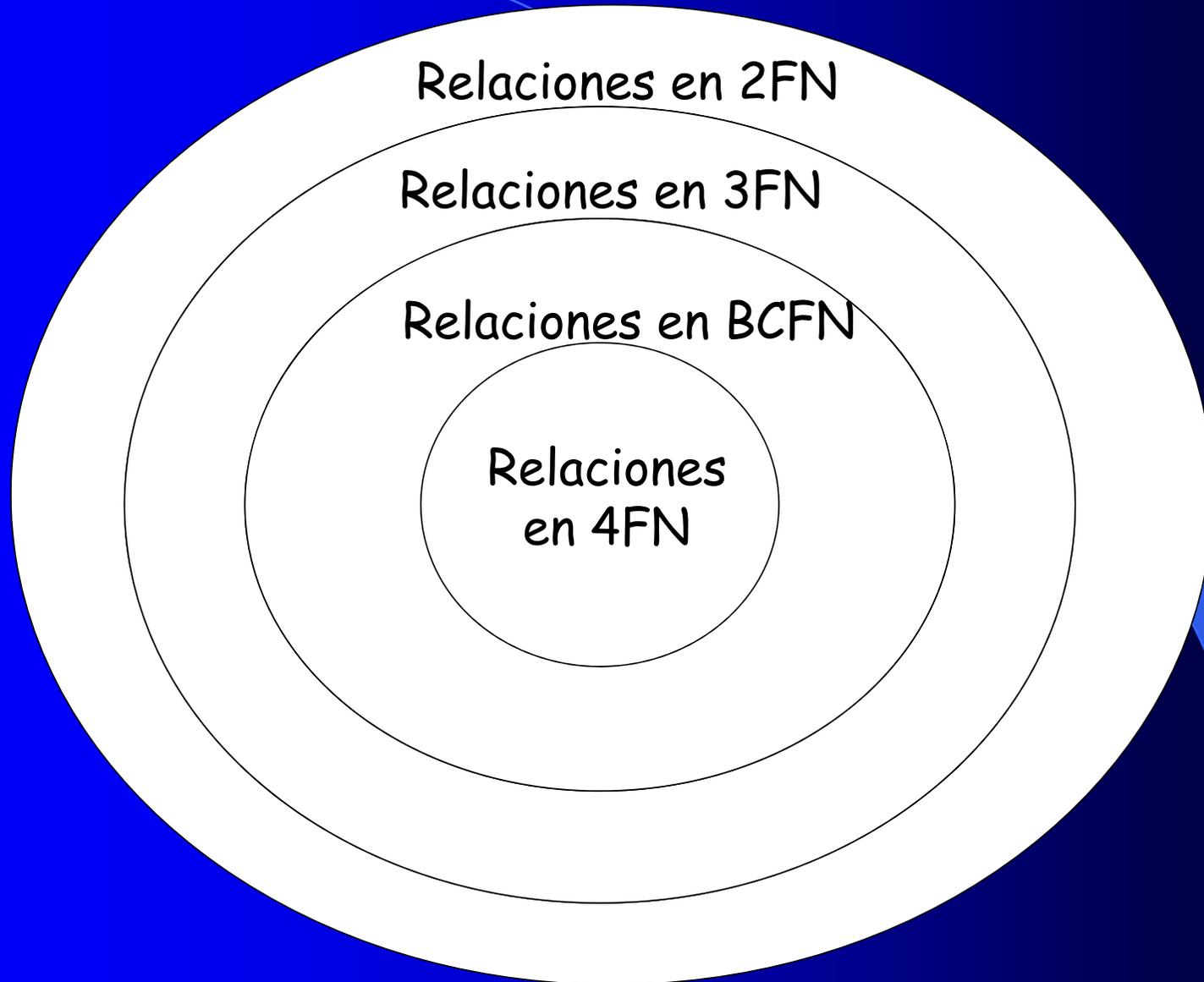
<u>Sno</u>	SName	SAddress	Position	Bno	Tel_No	Fax_No
S11	Jane Doe	11 Wood St.	Manager	B5	817-256-2234	817-256-2231
S23	Ann Martin	114 S. Main	Deputy	B4	972-456-8970	972-456-8842
S2	Leslie King	112 S. Main	Deputy	B4	972-456-8970	972-456-8842
S15	Matt Hoffa	29 Market St.	Assistant	B8	317-869-4511	317-869-1123
S45	Jill Emory	11 S. Elm	Manager	B4	972-456-8970	972-456-8842

- Clave primaria
- Nuevo empleado en sucursal B4
- Nueva sucursal
- Despiden al primer empleado
- Cambia el número de teléfono de B4

# Formas Normales



# Relación Formas Normales



# Conceptos Básicos

- Clave
- Clave candidata:
  - clave primaria
  - claves secundarias
- Clave extranjera (ajena)

# Dependencia Funcional

- $A \rightarrow B$ 
  - determinante
- Ejemplos:
  - ISBN  $\rightarrow$  BookTitle
  - EmpID, Course\_Title  $\rightarrow$  DateCompleted
  - SSN  $\rightarrow$  Name, Address, Birthdate
- $A \rightarrow R$  sii A es clave candidata
- Casos triviales (se excluyen):
  - B es subconjunto de A

# Obtención de Dependencias Funcionales

A	B
1	4
1	5
3	7

- ¿Cuáles son las dependencias funcionales?
- ¿Hay algo que se pueda deducir de los datos?
- De una instancia de una relación sólo pueden obtenerse contraejemplos

# Dependencias Funcionales: Reglas de Inferencia

Amstrong

- Reflexive:  $X \supseteq Y$ , then  $X \rightarrow Y$
- Augmentive:  $\{X \rightarrow Y\}$  gives  $X, Z \rightarrow Y, Z$
- Transitive:  $\{X \rightarrow Y, Y \rightarrow Z\}$  gives  $X \rightarrow Z$
- Decomposition:  $\{X \rightarrow Y, Z\}$  gives  $X \rightarrow Y$
- Union:  $\{X \rightarrow Y, X \rightarrow Z\}$  gives  $X \rightarrow Y, Z$
- Pseudo-Transitive:  
 $\{X \rightarrow Y; W, Y \rightarrow Z\}$  gives  $W, X \rightarrow Z$

## ● Cierre de F

# Primera Forma Normal (I)

- No atributos multivaluados

- Todas las relaciones

- Ejemplo:

- PROJECTS(PROJECT\_ID, HOURS)
- EMP\_PROJ(SSN, E\_NAME, PROJECTS)



- EMP\_PROJ(SSN, PROJECT\_ID)
- PROJ\_HOURS(PROJECT\_ID, HOURS)
- EMP (SSN, E\_NAME)

- Algunos prohíben atributos compuestos (ej: número de cuenta de 20 dígitos, una fecha)

# Primera Forma Normal (II)

- Cómo evitar atributos multivaluados:
  - en relación aparte con clave primaria la combinación
  - expandir la clave con el atributo multivaluado
  - sustituir por varios atributos
- Cómo evitar relaciones anidadas:
  - propagar la clave primaria



# ¿Está en 1FN? (I)

<u>Sno</u>	SName	SAddress	Position	Bno	Tel_No	Fax_No
S11	Jane Doe	11 Wood St.	Manager	B5	817-256-2234	817-256-2231
S23	Ann Martin	114 S. Main	Deputy	B4	972-456-8970	972-456-8842
⇒ S2	Leslie King	112 S. Main	Deputy	B4	972-456-8970	972-456-8842
S15	Matt Hoffa	29 Market St.	Assistant	B8	317-869-4511	317-869-1123
⇒ S2	Leslie King	112 S. Main	Deputy	B4	972-456-8970	972-456-8842

# ¿Está en 1FN? (II)

<u>Sno</u>	SName	SAddress	Position	Bno	Tel_No	Fax_No
S11	Jane Doe	11 Wood St.	Manager	B5	817-256-2234	817-256-2231
S23	Ann Martin	114 S. Main	Deputy, Assistant	B4	972-456-8970	972-456-8842
S2	Leslie King	112 S. Main	Deputy	B4	972-456-8970	972-456-8842
S15	Matt Hoffa	29 Market St.	Assistant	B8	317-869-4511	317-869-1123
S45	Jill Emory	11 S. Elm	Manager	B4	972-456-8970	972-456-8842

# Paso a Primera Forma Normal (I)

Orders

<b>Order Number</b>	<b>Order Date</b>	<b>Part Number</b>	<b>Number of Units</b>
12489	9/02/01	AX12	11
12491	9/02/01	BT04	1
		BZ66	1
12494	9/04/01	CB03	4

# Paso a Primera Forma Normal

(II)

Orders

<b>Order Number</b>	<b>Order Date</b>	<b>Part Number</b>	<b>Number of Units</b>
12489	9/02/01	AX12	11
12491	9/02/01	BT04	1
12491	9/02/01	BZ66	1
12494	9/04/01	CB03	4

# ¿Es 1FN Suficiente?

- Employee(EmpID, Name, DeptName, Salary, CourseTitle, DateCompleted)
- Problemas:
  - Inserción: insertar un empleado que no esté en ningún curso
  - Borrado: si borramos el último empleado que está en cierto curso
  - Modificación de los datos de un empleado
- Dependencias funcionales:
  - EmpID, CourseTitle → DateCompleted
  - EmpID → Name, DeptName, Salary

# Segunda Forma Normal (I)

- 1FN y no dependencias funcionales parciales
  - atributos no clave que dependen de parte de la clave
- 1FN equivale a 2FN si:
  - no hay atributos no claves ó
  - la clave es atómica
- Employee no está en 2FN:
  - EmpID → Name, DeptName, Salary

# Segunda Forma Normal (II)

- Cómo pasar a 2FN:
  - asociar los atributos implicados sólo con la parte de la clave de la que depende

# Paso a Segunda Forma Normal

<u>EmpID</u>	Name	DeptName	Salary
--------------	------	----------	--------

A red line connects the EmpID cell to the Name, DeptName, and Salary cells, with three red arrows pointing down to each of these three cells, indicating that EmpID functionally determines Name, DeptName, and Salary.

Dependencias  
funcionales  
completas

<u>EmpID</u>	<u>CourseTitle</u>	DateCompleted
--------------	--------------------	---------------

A red line connects the CourseTitle cell to the DateCompleted cell, with a red arrow pointing up to the DateCompleted cell, indicating that CourseTitle functionally determines DateCompleted.

# Ejemplo (I)

- Student: Student\_ID, Activity, Fee
- Clave: Student\_ID, Activity
- Dependencias funcionales: Activity → Fee

Student_ID	Activity	Fee
222-22-2020	Swimming	30
232-22-2111	Golf	100
222-22-2020	Golf	100
255-24-2332	Hiking	50

# Ejemplo (II)

## STUDENT\_ACTIVITY

<u>Student ID</u>	<u>Activity</u>
-------------------	-----------------

Clave: Student\_ID, Activity

## ACTIVITY\_COST

<u>Activity</u>	Fee
-----------------	-----

Clave: Activity

Activity → Fee

Student_ID	Activity
222-22-2020	Swimming
232-22-2111	Golf
222-22-2020	Golf
255-24-2332	Hiking

Activity	Fee
Swimming	30
Golf	100
Hiking	50

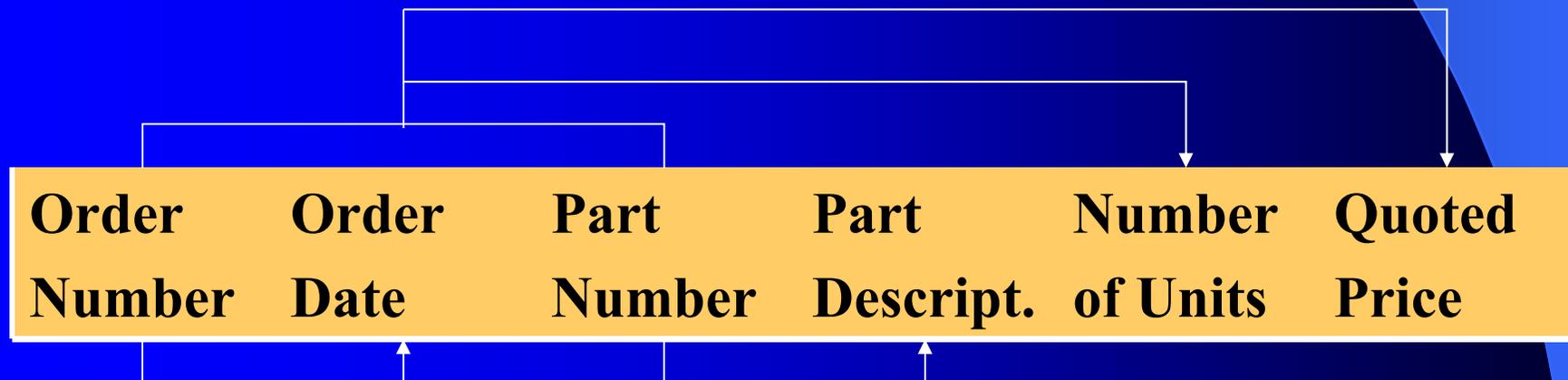
# Otro Ejemplo (I)

Orders

<b>Order Number</b>	<b>Order Date</b>	<b>Part Number</b>	<b>Part Descript.</b>	<b>Number of Units</b>	<b>Quoted Price</b>
12489	9/02/01	AX12	Iron	11	\$14.95
12491	9/02/01	BT04	Gas Grill	1	\$149.99
12491	9/02/01	BZ66	Washer	1	\$399.99
12494	9/04/01	CB03	Bike	4	\$279.99
12500	9/05/01	BT04	Gas Grill	1	\$149.99

# Otro Ejemplo (II)

- **Orders (Order Number, Order Date, Part Number, Part Description, Number of Units, Quoted Price)**
- **Dependencias funcionales:**
  - **Order Number → Order Date**
  - **Part Number → Part Description**
  - **Order Number, Part Number → Number of Units, Quoted Price**



# Otro Ejemplo (III)

Orders

Order Number	Order Date
12489	9/02/01
12491	9/02/01
12494	9/04/01
12500	9/05/01

Parts

Part Number	Part Descript.
AX12	Iron
BT04	Gas Grill
BZ66	Washer
CB03	Bike

Order Line

Order Number	Part Number	Number of Units	Quoted Price
12489	AX12	11	\$14.95
12491	BT04	1	\$149.99
12491	BZ66	1	\$399.99
12494	CB03	4	\$279.99
12500	BT04	1	\$149.99

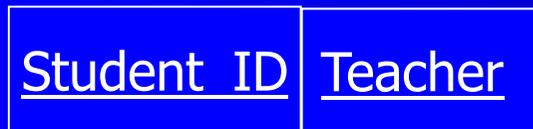
# Un Tercer Ejemplo (I)

- Student\_Teacher: Student\_ID, Subject, Teacher
- Clave: Student\_ID, Teacher
- Dependencias funcionales:
  - Teacher → Subject

<b>Student_ID</b>	<b>Subject</b>	<b>Teacher</b>
222-22-2020	Economy	Leigh
232-22-2111	Management	Gowan
222-22-2020	Economy	Roberts
222-22-2111	Marketing	Reynolds
255-24-2332	Marketing	Reynolds

# Un Tercer Ejemplo (II)

## STUDENT\_TEACHER



Key: Student\_ID, Teacher

## TEACHER\_SUBJECT



Key: Teacher

Teacher → Subject

Student_ID	Teacher
222-22-2020	Leigh
232-22-2111	Gowan
222-22-2020	Roberts
222-22-2111	Reynolds
255-24-2332	Reynolds

Teacher	Subject
Leigh	Economy
Gowan	Management
Roberts	Economy
Reynolds	Marketing

# ¿Es 2FN suficiente? (I)

- Customer(CustomerID, Name, Salesperson, Region)
- Dependencias funcionales:
  - CustomerID → Name, Salesperson
  - Salesperson → Region
- Problemas:
  - Inserción: insertar un vendedor que no tenga cliente
  - Borrado: si borramos el último cliente de cierto vendedor
  - Modificación: de la región de un vendedor
  - Redundancia: repetir la región cada vez que aparezca un vendedor

## ¿Es 2FN suficiente? (II)

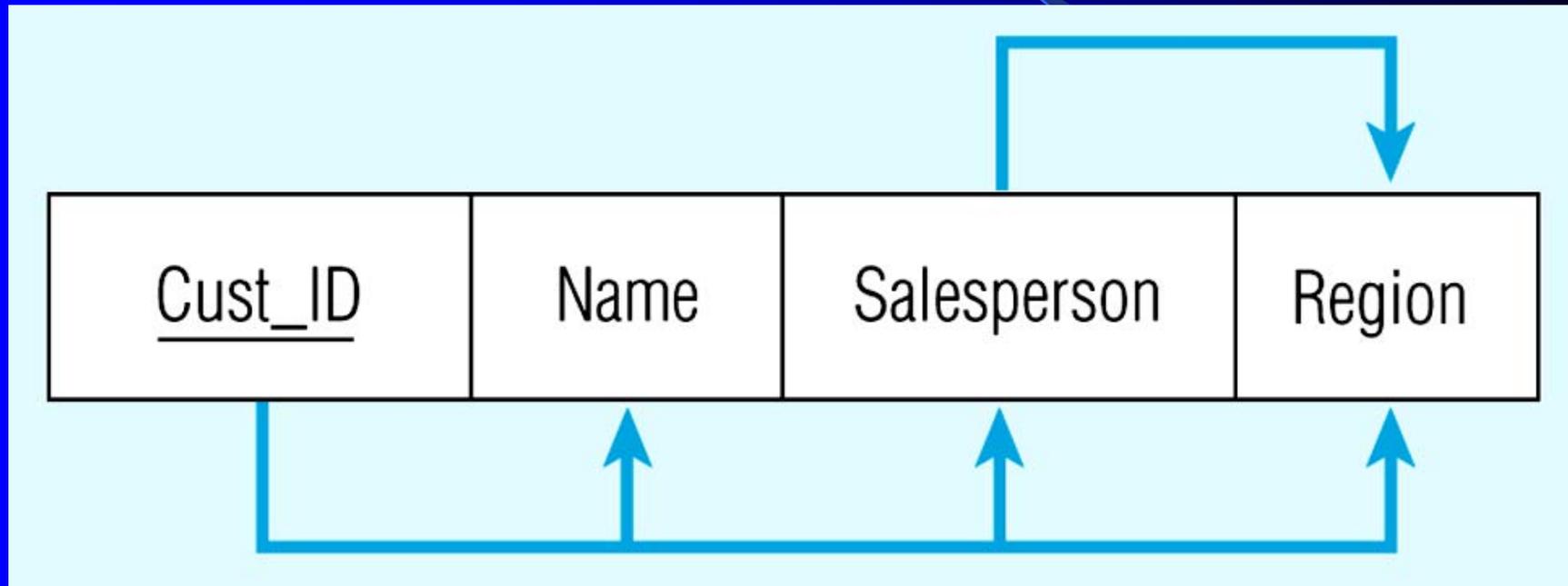
### SALES

Cust_ID	Name	Salesperson	Region
8023	Anderson	Smith	South
9167	Bancroft	Hicks	West
7924	Hobbs	Smith	South
6837	Tucker	Hernandez	East
8596	Eckersley	Hicks	West
7018	Arnold	Faulb	North

# Tercera Forma Normal

- 2FN y no dependencias transitivas
- Dependencia transitiva:
  - dependencia funcional entre atributos no clave
  - atributo no clave que depende indirectamente
  - dependencia más específica que la de la clave
- ¿Qué pasa con los atributos clave que dependen indirectamente de la clave?

# Ejemplo de Dependencia Transitiva



CustomerID → Salesperson → Region

↑  
Atributo no clave

# Paso a Tercera Forma Normal

SALES1			SPERSON	
Cust_ID	Name	Salesperson	Salesperson	Region
8023	Anderson	Smith	Smith	South
9167	Bancroft	Hicks	Hicks	West
7924	Hobbs	Smith	Hernandez	East
6837	Tucker	Hernandez	Faulb	North
8596	Eckersley	Hicks		
7018	Arnold	Faulb		

# Ejemplo (I)

- Student: Student\_ID, Building, Fee
- Clave: Student\_ID
- Dependencias funcionales:
  - Student\_ID → Building
  - Building → Fee

<b>Student_ID</b>	<b>Building</b>	<b>Fee</b>
222-22-2020	Dabney	1200
232-22-2111	Liles	1000
222-22-5554	The Range	1100
255-24-2332	Dabney	1200
330-25-7789	The Range	1100

# Ejemplo (II)

## STUDENT HOUSING



Clave: Student\_ID

Student\_ID → Building

## BUILDING COST



Clave: Building

Building → Fee

Student_ID	Building
222-22-2020	Dabney
232-22-2111	Liles
222-22-5554	The Range
255-24-2332	Dabney
330-25-7789	The Range

Building	Fee
Dabney	1200
Liles	1000
The Range	1100

# Otro Ejemplo (I)

Customer

<b>Customer Number</b>	<b>Cust Last Name</b>	<b>Cust First Name</b>	<b>Credit Balance</b>	<b>Credit Limit</b>	<b>Sales Rep Number</b>	<b>Slsr Last Name</b>	<b>Slsr First Name</b>
124	Adams	Sally	\$824.45	\$1000	03	Jones	Mary
256	Samuels	Ann	\$21.43	\$1500	06	Smith	William
311	Charles	Don	\$345.54	\$1000	12	Diaz	Miguel
315	Daniels	Tom	\$770.45	\$750	06	Smith	William
405	Williams	Al	\$450.56	\$1500	12	Diaz	Miguel

# Otro Ejemplo (II)

Customer

Customer Number	Cust Last Name	Cust First Name	Balance	Credit Limit	Sales Rep Number
124	Adams	Sally	\$824.45	\$1000	03
256	Samuels	Ann	\$21.43	\$1500	06
311	Charles	Don	\$345.54	\$1000	12
315	Daniels	Tom	\$770.45	\$750	06
405	Williams	Al	\$450.56	\$1500	12

Sales Rep

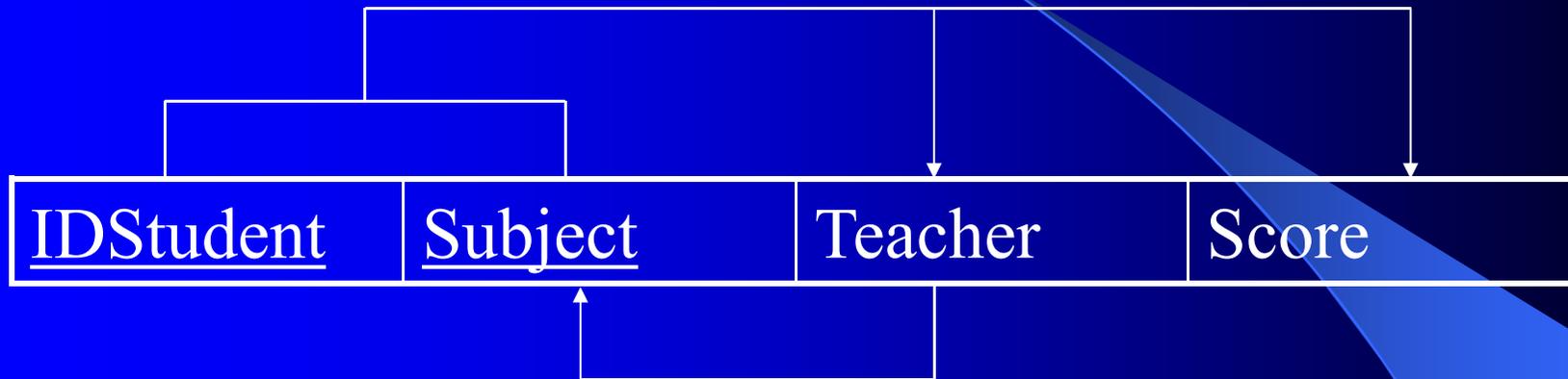
Sales Rep Number	Slsr Last Name	Slsr First Name
03	Jones	Mary
06	Smith	William
12	Diaz	Miguel

# ¿Es 3FN suficiente? (I)

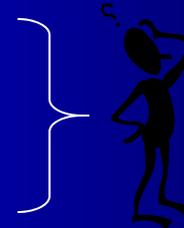
- Student(IDStudent, Subject, Teacher, Score)

<u>IDStudent</u>	<u>Subject</u>	Teacher	Score
123	Physics	Hawking	4.0
123	Music	Mahler	3.3
456	Lit	Michener	3.2
789	Music	Bach	3.7
678	Physics	Hawking	3.5

# ¿Es 3FN suficiente? (II)



- ¿En 1FN? —————> Sí
- ¿En 2FN? —————> Sí
- ¿En 3FN? —————> Sí
- Teacher → Subject
  - no es dependencia transitiva



## ¿Es 3FN suficiente? (III)

IDStudent	Subject	Teacher	Score
123	Physics	Hawking	4.0
123	Music	Mahler	3.3
456	Lit	Michener	3.2
789	Music	Bach	3.7
678	Physics	Hawking	3.5

- Cambio del profesor de Física
- Inserción de un profesor de Economía
- Borrado del estudiante 789

} Problemas

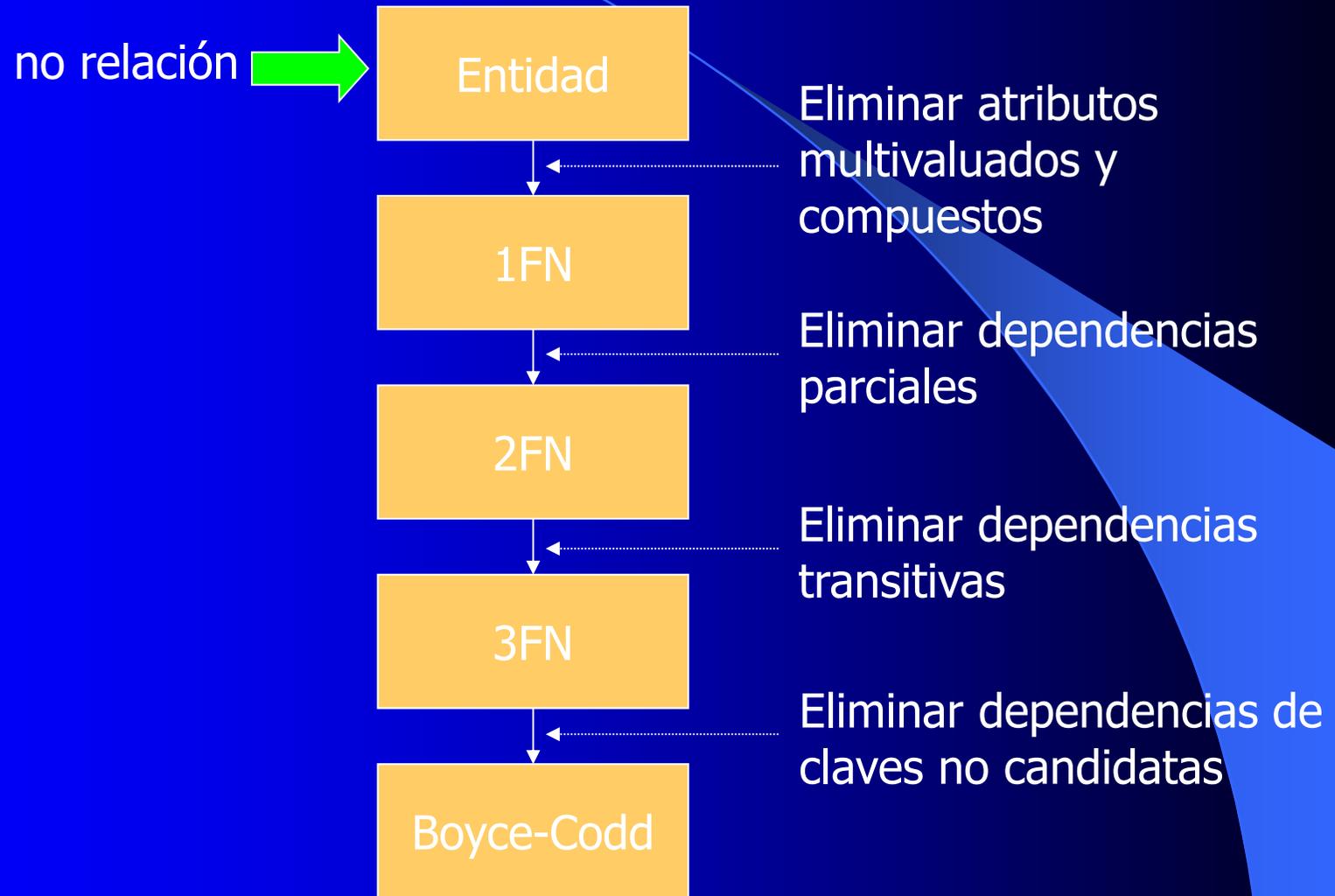
# Forma Normal de Boyce-Codd

- Todo determinante de dependencias funcionales debe ser clave
- ¿Por qué no es la 4FN?
- Si un atributo no contribuye a la descripción de una clave, colocarlo en otra relación

<u>IDStudent</u>	<u>Teacher</u>	Score
------------------	----------------	-------

<u>Teacher</u>	Subject
----------------	---------

# Proceso de Normalización



# Ejemplo de Normalización (I)

## Atributos

Puppy Number

Puppy Name

Kennel Code

Kennel Name

Kennel Location

Trick ID 1...n

Trick Name 1...n

Trick Where Learned 1...n

Skill Level 1...n



# Ejemplo de Normalización (II)

1FN

- Una relación para cada grupo de atributos relacionados
- Dar a cada relación una clave primaria
- Evitar atributos multivaluados



## Puppy Table

*Puppy Number (PK)*  
Puppy Name  
Kennel Code  
Kennel Name  
Kennel Location

## Trick Table

*Puppy Number (PK)*  
*Trick ID (PK)*  
Trick Name  
Trick Where Learned  
Skill Level

# Ejemplo de Normalización (III)

- Eliminar dependencias parciales:

– TrickID → Trick Name

Trick Table				
<i>Puppy #</i>	<i>Trick ID</i>	Trick Name	Where Learned	Skill Level
52	27	Roll Over	16	9
53	16	Nose Stand	9	9
54	27	Roll Over	9	5

Tricks

*Trick ID (PK)*  
Trick Name

Puppy Tricks

*Puppy Number (PK)*  
*Trick ID (PK)*  
Trick Where Learned  
Skill Level

Puppy Table

*Puppy Number (PK)*  
Puppy Name  
Kennel Code  
Kennel Name  
Kennel Location

2FN

# Ejemplo de Normalización (IV)

Puppy Table
<i>Puppy Number (PK)</i>
Puppy Name
Kennel Code
Kennel Name
Kennel Location



Puppies
<i>Puppy Number (PK)</i>
Puppy Name
Kennel Code

Kennels
<i>Kennel Code (PK)</i>
Kennel Name
Kennel Location

Tricks
<i>Trick ID (PK)</i>
Trick Name

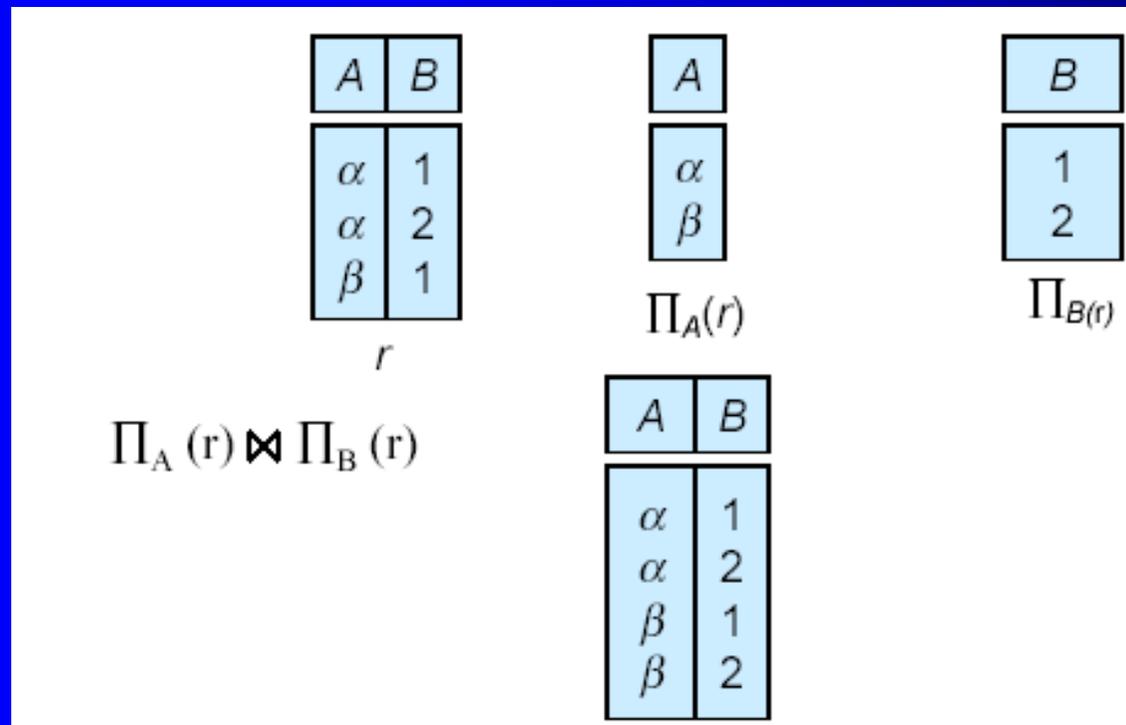


Puppy Tricks
<i>Puppy Number (PK)</i>
<i>Trick ID (PK)</i>
Trick Where Learned
Skill Level

- Todo determinante debe ser clave candidata:
  - Puppy Number →  
Puppy Name
  - Kennel Code →  
Kennel Name,  
Kennel Location

# Objetivos de Diseño (I)

- 1) Descomposición si pérdida



$$R_1 \cap R_2 \rightarrow R_1$$

$$R_1 \cap R_2 \rightarrow R_2$$

# Objetivos de Diseño (II)

- Ejemplo:
  - EMP\_PROJ(SSN, PNUMBER, HOURS, ENAME, PNAME, PLOCATION)
  - descompuesto en:
    - EMP\_LOCS(ENAME, PLOCATION)
    - EMP\_PROJ1(SSN, PNUMBER, HOURS, PNAME, PLOCATION)
- Problema: tuplas espúreas

# Objetivos de Diseño (III)

- 2) Conservación de dependencias:
  - evitar joins para comprobar dependencias funcionales
- 3) Evitar redundancias (formas normales):
  - desperdicio de espacio
  - inconsistencias

# Ejemplo

●  $R=(A, B, C)$ :

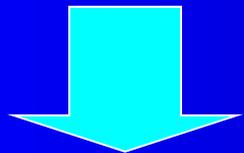
–  $A \rightarrow B$

–  $B \rightarrow C$

a)  $R1(A, B), R2(B, C)$



b)  $R1(A, B), R2(A, C)$



Con b) no se conserva la dependencia  $B \rightarrow C$

# Otro Ejemplo

- Banquero(NSUCURSAL, NCLIENTE, NBANQUERO)
  - NBANQUERO → NSUCURSAL (no está en FNBC)
  - NSUCURSAL, NCLIENTE → NBANQUERO

## Otro Ejemplo (II)

SucursalBanquero(NBANQUERO, NSUCURSAL)

ClienteBanquero(NCLIENTE , NBANQUERO)

<u>NBANQUERO</u>	NSUCURSAL
1	1
2	1
3	3
4	3

<u>NCLIENTE</u>	<u>NBANQUERO</u>
1	3
1	4

No se conserva: NSUCURSAL, NCLIENTE → NBANQUERO

# BCFN y Preservación de Dependencias

- A veces, no puede obtenerse BCFN y conservar las dependencias al mismo tiempo (ver ejemplo anterior)
- Aunque siempre se puede obtener una descomposición 3FN sin pérdidas

# BCFN y Preservación de Dependencias: Ejemplo

- $R=(A, B, C)$ :
  - $F = \{AB \rightarrow C, C \rightarrow B\}$
- Claves candidatas:  $(A, B), (A, C)$
- No está en BCFN  $(C \rightarrow B)$
- Al descomponer se perderá  $AB \rightarrow C$

# De Ahí la Utilidad de 3FN...

- Siempre se puede obtener una descomposición en 3FN:
  - sin pérdidas (conservar junta una clave candidata)
  - conservando las dependencias
- Coste: algo de redundancia
- **BCFN**: todo atributo depende completamente de la clave
- **3FN**: todo atributo **no clave** depende completamente de la clave

# ¿Es suficiente con BCFN? (I)

## Stars

<i>name</i>	<i>street</i>	<i>city</i>	<i>title</i>
C. Fisher	123 Maple Str.	Hollywood	Star Wars
C. Fisher	5 Locust Ln.	Malibu	Star Wars
C. Fisher	123 Maple Str.	Hollywood	Empire Strikes Back
C. Fisher	5 Locust Ln.	Malibu	Empire Strikes Back
C. Fisher	123 Maple Str.	Hollywood	Return of the Jedi
C. Fisher	5 Locust Ln.	Malibu	Return of the Jedi

# ¿Es suficiente con BCFN? (II)

- Hay que asociar todas las direcciones con todas las películas
  - Redundancia (información de direcciones)
- Problemas
- ¿En BCFN? → Sí (todo es clave)

# Dependencias Multivaluadas

- $A \twoheadrightarrow B$
- Generalización de la dependencia funcional
- Declaración de la independencia entre conjuntos de atributos:
  - relación entre A y B independiente de entre A y R-B
- Si  $A \twoheadrightarrow B$ , entonces  $A \twoheadrightarrow R-B$
- Casos triviales (se excluyen):
  - B subconjunto de A
  - $A \cup B = R$

# Dependencias Multivaluadas: Algunas Reglas de Inferencia

- Transitividad:

- Si  $A \twoheadrightarrow B$  y  $A \twoheadrightarrow C$ , entonces  $A \twoheadrightarrow C$

- Complementariedad:

- Si  $A \twoheadrightarrow B$ , entonces  $A \twoheadrightarrow R - (A \cup B)$

- Unión:

- Si  $A \twoheadrightarrow B$  y  $A \twoheadrightarrow C$ , entonces  $A \twoheadrightarrow B, C$

# Cuarta Forma Normal

- Todo determinante de dependencias multivaluadas debe ser clave
- Mientras haya dependencias multivaluadas
  - descomponer la relación en dos relaciones
    - R1: Determinante, atributos determinados
    - R2: Determinante, atributos que no están en R1
  - para cada descomposición, comprobar nuevas dependencias

# Ejemplo (I)

## Stars

<i>name</i>	<i>street</i>	<i>city</i>	<i>title</i>
C. Fisher	123 Maple Str.	Hollywood	Star Wars
C. Fisher	5 Locust Ln.	Malibu	Star Wars
C. Fisher	123 Maple Str.	Hollywood	Empire Strikes Back
C. Fisher	5 Locust Ln.	Malibu	Empire Strikes Back
C. Fisher	123 Maple Str.	Hollywood	Return of the Jedi
C. Fisher	5 Locust Ln.	Malibu	Return of the Jedi

# Ejemplo (I)

## Stars

<i>name</i>	<i>street</i>	<i>city</i>	<i>title</i>
C. Fisher	123 Maple Str.	Hollywood	Star Wars
C. Fisher	5 Locust Ln.	Malibu	Star Wars
C. Fisher	123 Maple Str.	Hollywood	Empire Strikes Back
C. Fisher	5 Locust Ln.	Malibu	Empire Strikes Back
C. Fisher	123 Maple Str.	Hollywood	Return of the Jedi
C. Fisher	5 Locust Ln.	Malibu	Return of the Jedi

## Ejemplo (II)

- name → → street, city
- Hay que repetir:
  - cada dirección para cada película
  - cada película para cada dirección

Hechos  
independientes

<i>name</i>	<i>street</i>	<i>city</i>	<i>title</i>
C. Fisher	123 Maple Str.	Hollywood	Star Wars
C. Fisher	5 Locust Ln.	Malibu	Empire Strikes Back
C. Fisher	123 Maple Str.	Hollywood	Return of the Jedi

EXTENSIÓN  
INCOMPLETA

# Ejemplo (III)

- Descomponer en:
  - R1(name, street, city)
  - R2(name, title)

# Otro Ejemplo (I)

<i>course</i>	<i>teacher</i>	<i>book</i>
database	Hank	DB Concepts
database	Hank	Ullman
database	Sudarshan	DB Concepts
database	Sudarshan	Ullman
operating systems	Hank	OS Concepts
operating systems	Hank	Shaw
operating systems	Jim	OS Concepts
operating systems	Jim	Shaw

- Todos los atributos son clave  $\Rightarrow$  BCFN
- Anomalías:
  - insertar un nuevo profesor de bases de datos

# Otro Ejemplo (II)

<i>course</i>	<i>teacher</i>
database	Hank
database	Sudarshan
operating systems	Hank
operating systems	Jim

*teaches*

<i>course</i>	<i>book</i>
database	DB Concepts
database	Ullman
operating systems	OS Concepts
operating systems	Shaw

*text*

- *course* → → *teacher*
- *course* → → *book*

# Ejercicio 1

- Normalizar a 4FN la relación  $R(A,B,C,D)$ , donde:
  - $A \twoheadrightarrow B$
  - $A \twoheadrightarrow C$
- Clave: A, B, C, D (no dependencias funcionales)
- $A \twoheadrightarrow B$  y  $A \twoheadrightarrow C$  violan 4FN
- $R_1(A, B), R_2(A, C, D)$
- $A \twoheadrightarrow C$  viola R2:
  - $R_{21}(A, C), R_{22}(A, D)$

## Ejercicio 2

- $R(A,B,C)$ 
  - $A \twoheadrightarrow B$
  - $(a,b_1,c_1)$ ,  $(a,b_2,c_2)$ , y  $(a,b_3,c_3)$  son tuplas de  $R$
  - ¿Qué tuplas sabemos que deben estar en  $R$ ?
- Pista:  $A$  determina los valores de  $B$  con independencia de  $C$  (para cualquier valor de  $C$ )
- Respuesta: todas las tuplas de la forma  $(a, b, c)$  con  $b=b_1,b_2,b_3$  y  $c=c_1,c_2,c_3$  (9 tuplas)

# Más Formas Normales

- 5FN = Forma normal de proyección-join
  - generalización de las dependencias multivaluadas: dependencias de join
- Restricciones más generales llevan a la forma normal de Dominio/Clave
- Problemas de estas formas normales:
  - es difícil razonar con ellas
  - no hay un conjunto de reglas de inferencia completo y correcto

raramente  
usadas

# Nota al Margen

- Cuando se realiza diseño por síntesis, lo que se hace es deducir relaciones entre atributos a partir de las dependencias existentes
- El diseño por síntesis, sin embargo, no goza de mucha popularidad, entre otras cosas por el elevado número de dependencias que pueden estar implicadas

# Presentación Disponible en...

<http://webdiis.unizar.es/~silarri/>

2 Noviembre, 2005 - Sergio Ilarri

FIN

2 Noviembre, 2005 - Sergio Ilarri