

# UN MODELO RELACIONAL DE DATOS PARA GRANDES BANCOS DE DATOS COMPARTIDOS

E. F. CODD  
LABORATORIO DE INVESTIGACIÓN DE IBM, SAN JOSÉ, CALIFORNIA

POR MORENO DEL CASTILLO ARACELI  
SEMESTRE 2021-1

# PROBLEMÁTICA QUE ANALIZÓ E.F. CODD

PROTECCIÓN DE LA INFORMACIÓN

ORGANIZACIÓN DE DATOS (REPRESENTACIÓN INTERNA)

EVITAR AFECTACIONES CON CAMBIOS INTERNOS DE LOS DATOS  
INCLUSO CON CAMBIOS EXTERNOS

CAMBIOS EN LA REPRESENTACIÓN DE DATO ERAN NECESARIOS  
COMO RESULTADO DE CAMBIOS EN LA CONSULTA,  
ACTUALIZACIÓN E INFORME DE TRÁFICO Y CRECIMIENTO  
NATURAL DEL TIPO DE INFORMACIÓN ALMACENADA.

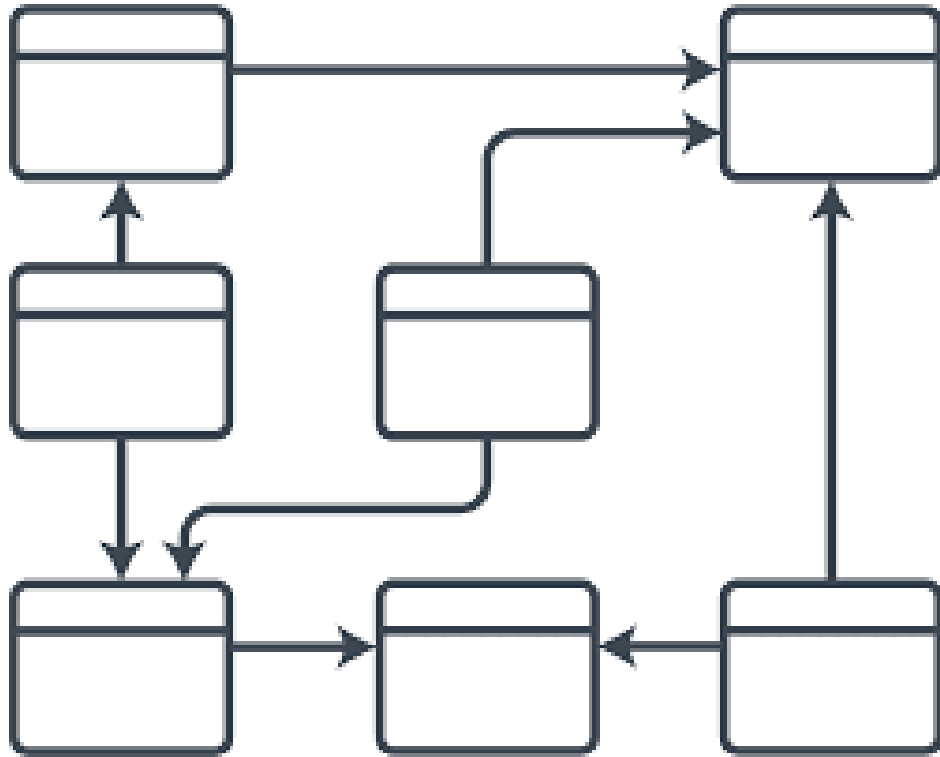
SISTEMAS DE DATOS FOMENTADOS EXISTENTES PROPORCIONAN  
A LOS USUARIOS ARCHIVOS ESTRUCTURADOS EN UN ÁRBOL O  
RED O DE FORMA MÁS GENERAL MODELOS DE DATOS.

DEFICIENCIAS DE LOS MODELOS EXISTENTES.

DEPENDENCIA DE LA INFORMACIÓN.







## PROPUESTAS DE E.F. CODD

PRIMERA VEZ QUE ALGUIEN ESPECIFICÓ UN MODELO ABSTRACTO PARA LA REPRESENTACIÓN DE DATOS.

SE BUSCABA UN MODELO DE DATOS QUE NO DEPENDA DE COMO SE ORGANIZAN LOS DATOS EN LA MÁQUINA

UN MODELO QUE FACILITE LAS BÚSQUEDAS, CONSULTAS Y ACTUALIZACIONES DE LOS DATOS SIN CAMBIAR LA MANERA EN QUE FÍSICAMENTE SE DISTRIBUYE

# DEPENDENCIAS DE DATOS EN LOS SISTEMAS

CODD NOS SEÑALA 3 TIPOS DE  
DEPENDENCIA BÁSICOS:

- DEPENDENCIA DE ORDEN
- DEPENDENCIA DE INDEXACIÓN
- DEPENDENCIA DE LA RUTA DE ACCESO





# DEPENDENCIA DE ORDEN

DEPENDE DEL ORDEN FÍSICO DE LOS  
DATOS.

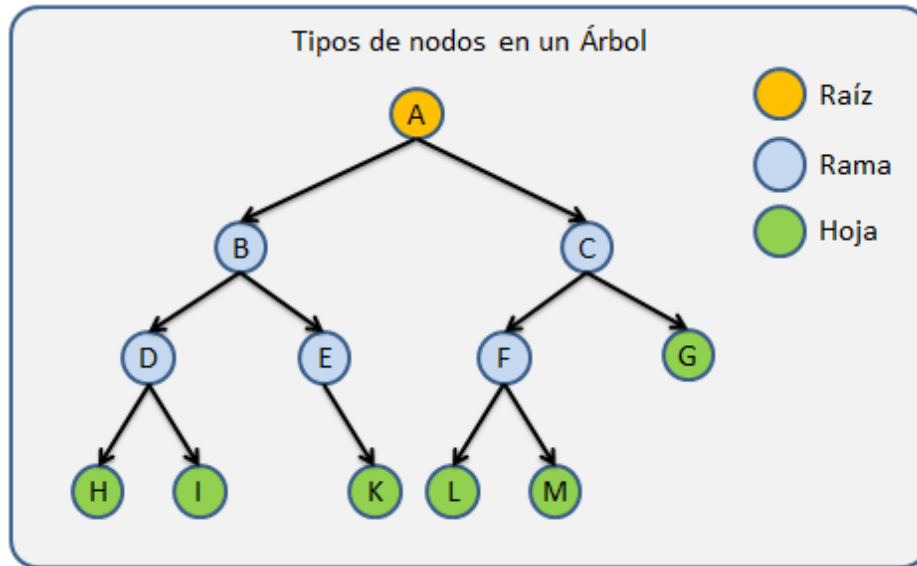
## DEPENDENCIA DE INDEXACIÓN

UN ÍNDICE SE SUELE CONSIDERARSE COMO UN COMPONENTE ORIENTADO AL RENDIMIENTO DE LA REPRESENTACIÓN DE DATOS. TIENDE A MEJORAR LA RESPUESTA A CONSULTAS Y ACTUALIZACIONES Y, AL MISMO TIEMPO, RALENTIZAR LA RESPUESTA A LAS INSERCIONES Y ELIMINACIONES.

DESDE UN PUNTO DE VISTA INFORMATIVO, UN ÍNDICE ES UN COMPONENTE REDUNDANTE DE LA REPRESENTACIÓN DE DATOS, LO CUAL GENERA UN PROBLEMA, ADEMÁS QUE CON UN CAMBIO DE INDEXACIÓN ESTOS NO FUNCIONARÍAN CORRECTAMENTE.



# DEPENDENCIA DE LA RUTA DE ACCESO



DEPENDEN DEL ORDEN DE LOS DATOS, GENERALMENTE ESTOS SE ACOMODABAN EN ÁRBOLES O GRAFOS, LOS CUALES ERAN PROGRAMADOS PARA ACCESAR A ELLOS DE UNA MISMA MANERA, AL CAMBIAR EL ORDEN DE ESTOS, POR ENDE EL PROGRAMA TENIA QUE CAMBIAR, ES POR ESO QUE ERA MUY DIFÍCIL CAMBIAR EL ORDEN DE LOS ÁRBOLES O GRAFOS SIN ALTERAR LOS PROGRAMAS.



# VISTA RELACIONAL DE DATOS

SE BASA EN RELACIONES CONOCIDAS POR LA TEORÍA DONDE SE BASA EN UN SET DE N TUPLAS, DONDE EL PRIMER ELEMENTO DE LA TUPLA ES TOMADO POR EL PRIMERO DEL SEGUNDO ELEMENTO, LO QUE ES FÁCIL REPRESENTAR EN UNA TUPLA DE DOS DIMENSIONES. DONDE LA FILA ES UNA TUPLA Y LA COLUMNA ES UN ELEMENTO DEL CONJUNTO EN PARTICULAR.

Campos o Atributos

Registros o Tuplas

Nombre	Nacionalidad	Institución
Sofia	España	O.N.U.
John	EE. UU.	O.M.S.
Jacob	Francia	N.A.S.A.
Sebastian	Chila	UNASUR

# VISTA RELACIONAL DE DATOS

## PROPIEDADES:

- CADA FILA REPRESENTA UNA N-TUPLA DE R.
- EL ORDEN DE LAS FILAS ES IRRELEVANTE.
- TODAS LAS FILAS SON DISTINTAS.
- EL ORDEN DE LAS COLUMNAS ES SIGNIFICATIVO: CORRESPONDE AL ORDEN  $S_1, S_2, \dots, S_n$  DE LOS DOMINIOS EN LOS QUE SE DEFINE R
- LA IMPORTANCIA DE CADA COLUMNA SE TRANSMITE PARCIALMENTE ETIQUETÁNDOLA CON EL NOMBRE DEL DOMINIO CORRESPONDIENTE.

<i>supply</i>	<i>(supplier</i>	<i>part</i>	<i>project</i>	<i>quantity)</i>
1	1	2	5	17
1	1	3	5	23
2	2	3	7	9
2	2	7	5	4
4	4	1	1	12

FIG. 1. A relation of degree 4

<i>supply</i>	<i>(supplier</i>	<i>part</i>	<i>project</i>	<i>quantity)</i>
1	1	2	5	17
1	1	3	5	23
2	2	3	7	9
2	2	7	5	4
4	4	1	1	12

FIG. 1. A relation of degree 4



# VISTA RELACIONAL DE DATOS

LOS DATOS EN UNA BASE DE DATOS PUEDEN VERSE COMO UN CONJUNTO DE RELACIONES QUE VARÍAN CON EL TIEMPO, YA QUE SE INSERTAN BORRAN Y SE MODIFICAN.

LAS RELACIONES SON LAS CONTRAPARTES DESORDENADAS DE LAS RELACIONES.

ES POR ELLO QUE CODD SIENTA LAS BASES DE LOS CONCEPTOS MODERNOS, YA QUE EL DOMINIO ES LO MISMO QUE LA COLUMNA Y UNA RELACIÓN ES EL EQUIVALENTE A UNA TABLA

# VISTA RELACIONAL DE DATOS

PRIMARY KEY ES EL DOMINIO QUE IDENTIFICA DE FORMA ÚNICA A LA RELACIÓN Y PUEDE SER UN SOLO DOMINIO O UNA COMBINACIÓN DE VARIOS DOMINIOS.

FOREING KEY ES LA LLAVE PRIMARIA DE OTRA TABLA REPRESENTADA EN UNA TABLA, LO CUAL EMPIEZA A ESTABLECER LAS RELACIONES ENTRE LAS TABLAS, ES DECIR REFERENCIA ELEMENTOS DE OTRAS TABLAS



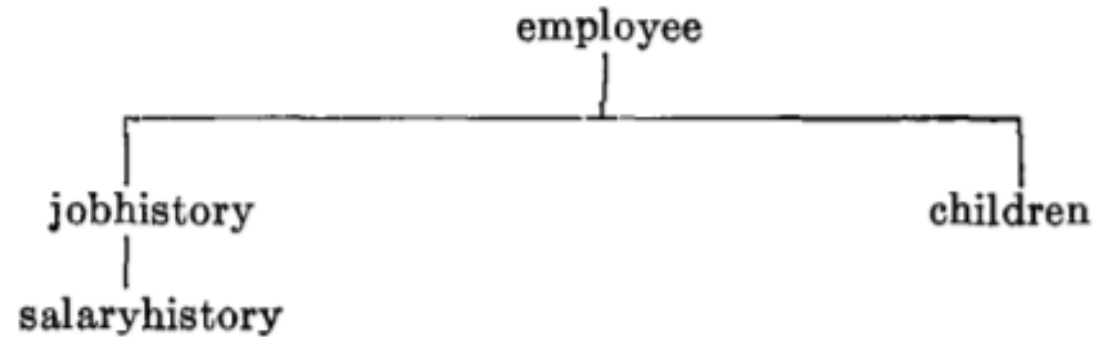
# VISTA RELACIONAL DE DATOS

CADA DOMINIO NO NECESITA UN DOMINIO ATÓMICO SIMPLE, PODRÍA TENER VALORES COMPLEJOS NO ATÓMICOS QUE SON PARTE DE UNA COLUMNA.

TENEMOS LA POSIBILIDAD DE LA EXISTENCIA DE TABLAS ANIDADAS

# FORMA NORMAL

CONSISTE EN TOMAR UN  
CONJUNTO DE RELACIONES  
DE ANIDADAS O NO  
ATÓMICAS Y LUEGO  
NORMALIZARLAS PARA  
GENERAR TABLAS PLANAS,  
QUE SOLO TIENEN DATOS  
ATÓMICOS



employee (*man#*, name, birthdate, jobhistory, children)  
jobhistory (*jobdate*, title, salaryhistory)  
salaryhistory (*salarydate*, salary)  
children (*childname*, birthyear)

FIG. 3(a). Unnormalized set

employee' (*man#*, name, birthdate)  
jobhistory' (*man#*, *jobdate*, title)  
salaryhistory' (*man#*, *jobdate*, *salarydate*, salary)  
children' (*man#*, *childname*, birthyear)

FIG. 3(b). Normalized set



# FORMA NORMAL



Razón por la que se busca normalizar las relaciones es porque hace que sea mucho más fácil almacenarlas y también transferirlas entre sistemas.



Además que facilita la búsqueda de información así como la modificación de la misma.

# Normalización

Código	Nombre	Apellido	Dormitorio
A-001	Albert	Einstein	A-305
A-002	Edgar	Codd	B-102

Alumno	Curso
A-001	Matemática I
A-001	Modelo de Datos
A-001	Física I
A-002	Modelo de Datos
A-002	Química I

## Son Relaciones:

- No hay tuplas repetidas
- No importa el orden
- Hay una Llave Primaria
- Atributos atómicos

No hay Grupos Repetitivos

Primera Forma Normal



## ASPECTOS LINGÜÍSTICOS

PERMITE EL DESARROLLO DE UN SUBLENGUAJE DE DATOS UNIVERSAL BASADO EN UN CÁLCULO DE PREDICADOS APLICADO.

LA UNIVERSALIDAD DEL SUBLENGUAJE DE DATOS RADICA EN SU HABILIDAD DESCRIPTIVA.

ESTE LENGUAJE DE DATOS ES QUE NO ESTÁ DEFINIDO EN TÉRMINOS DE EJECUCIÓN.

PROPONE UN LENGUAJE FAMILIAR Y FÁCIL DE EMPLEAR ENFOCADO EN LA FORMA DE LAS TABLAS Y COLUMNAS



## RELACIONES EXPRESIBLES, NOMBRADAS Y ALMACENADAS

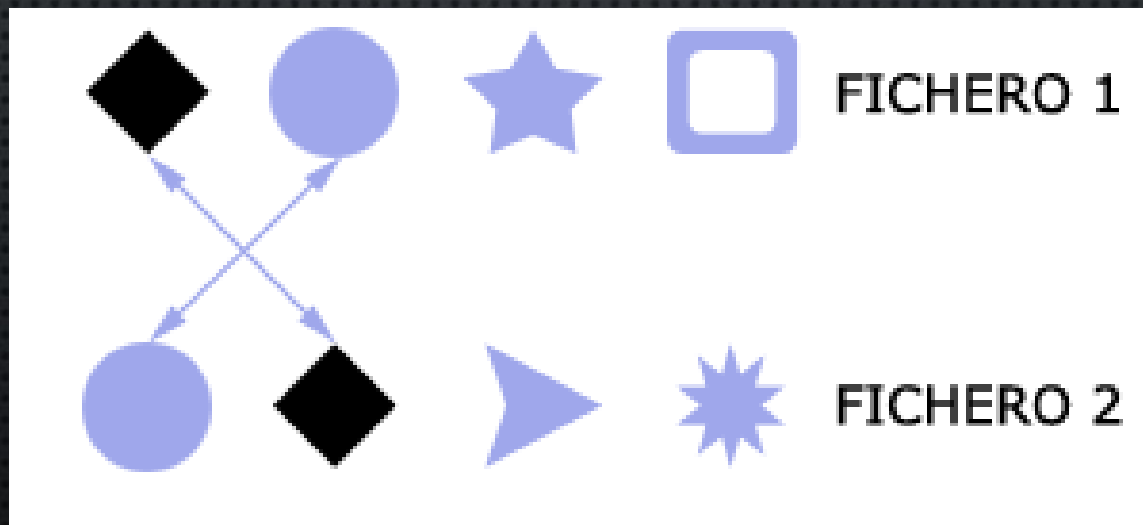
- EL CONJUNTO NOMBRADO ES EL CONJUNTO DE TODAS AQUELLAS RELACIONES QUE LA COMUNIDAD DE LOS USUARIOS PUEDE IDENTIFICARSE MEDIANTE UN SIMPLE NOMBRE (IDENTIFICADOR).
- EL CONJUNTO EXPRESABLE ES LA COLECCIÓN TOTAL DE RELACIONES QUE SE PUEDE DESIGNAR MEDIANTE EXPRESIONES EN EL LENGUAJE DE DATOS. LAS EXPRESIONES SE CONSTRUYEN A PARTIR DE NOMBRES SIMPLES DE RELACIONES EN EL CONJUNTO NOMBRADO; NOMBRES DE GENERACIONES, ROLES Y DOMINIOS; CONECTIVOS LÓGICOS; LOS CUANTIFICADORES DE LOS CÁLCULOS DE PREDICADOS; Y CIERTOS SÍMBOLOS DE RELACIÓN CONSTANTE COMO  $\sim$ ,  $=$ ,  $>$ .

# RELACIONES EXPRESIBLES, NOMBRADAS Y ALMACENADAS

- EL CONJUNTO NOMBRADO ES UN SUBCONJUNTO DEL CONJUNTO EXPRESABLE, GENERALMENTE UN SUBCONJUNTO MUY PEQUEÑO. ASOCIAR CON EL CONJUNTO NOMBRADO UNA COLECCIÓN DE DECLARACIONES QUE DEFINEN ESTAS RESTRICCIONES INDEPENDIENTES DEL TIEMPO.



# REDUNDANCIA Y COHERENCIA



# OPERACIONES SOBRE RELACIONES

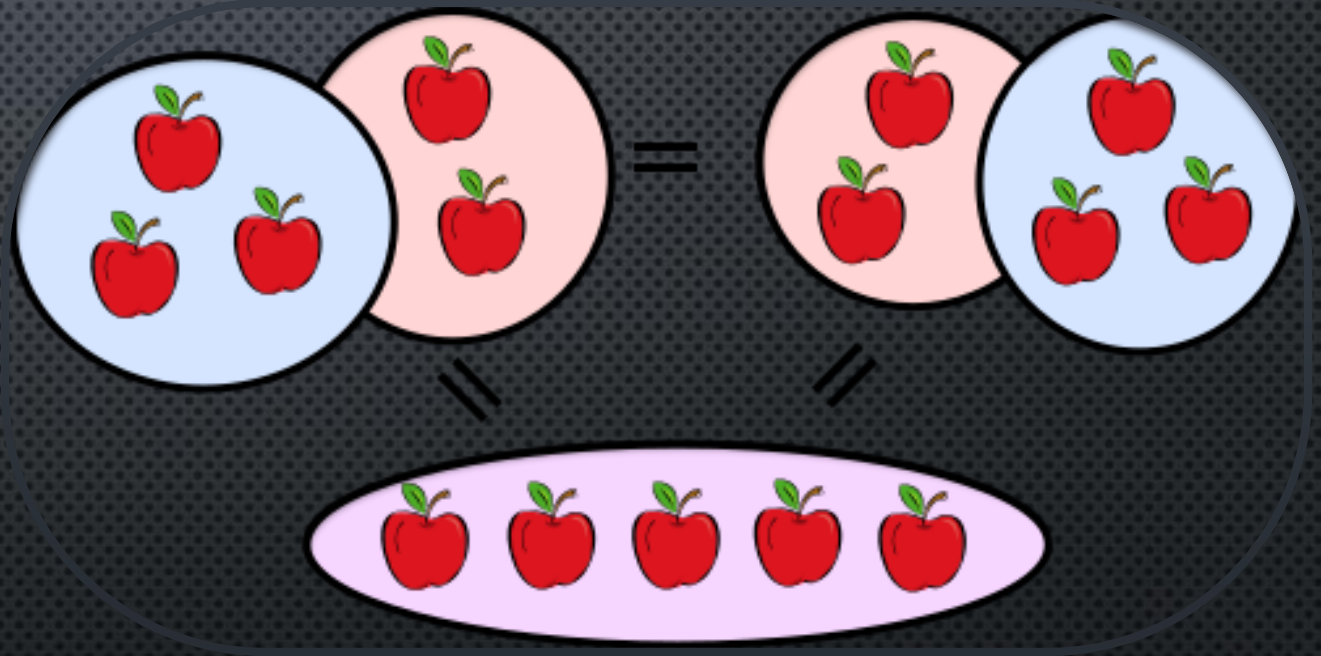
- DADO QUE LAS RELACIONES SON CONJUNTOS, TODAS LAS OPERACIONES HABITUALES DE CONJUNTOS SON APLICABLE A ELLOS, AUNQUE EL RESULTADO PUEDE NO SER UN RELACIÓN; POR EJEMPLO, LA UNIÓN DE UNA RELACIÓN BINARIA Y UNA RELACIÓN TERNARIA NO ES UNA RELACIÓN.



# PERMUTACIÓN.

LA PRIMERA ES LA IMITACIÓN Y UNA PERMUTACIÓN DE LA RELACIÓN ES SIMPLEMENTE OTRA RELACIÓN CON LAS COLUMNAS CONMUTADAS, LAS CUALES GENERALMENTE NO AFECTAN AL DESEMPEÑO.

EL ÚNICO MOMENTO EN QUE PODRÍA SER RELEVANTE ES SI ESTÁN TRATANDO DE OPTIMIZAR PARA EL PERFORMANCE.





# PROYECCIÓN

- SE SELECCIONAN CIERTAS COLUMNAS DE UNA RELACIÓN (TACHANDO LAS OTRAS) Y LUEGO ELIMINAMOS DE LA MATRIZ RESULTANTE CUALQUIER DUPLICACIÓN EN LAS FILAS, LA MATRIZ FINAL REPRESENTA UNA RELACIÓN QUE SE DICE QUE ES UNA PROYECCIÓN DE LA RELACIÓN DADA.

• Relación r:

A	B	C
$\alpha$	10	1
$\alpha$	20	1
$\beta$	30	1
$\beta$	40	2

$\Pi_{A,C}(r)$

A	C
$\alpha$	1
$\alpha$	1
$\beta$	1
$\beta$	2

=

A	C
$\alpha$	1
$\beta$	1
$\beta$	2

# JOIN

$$R * S = \{ (a, b, c) : R(a, b) \wedge S(b, c) \}$$

ES LA UNIÓN DE DOS TABLAS O DOS RELACIONES

SUPONGAMOS QUE SE NOS DAN DOS RELACIONES BINARIAS, QUE TIENEN ALGÚN DOMINIO EN COMÚN, ESTAS SE PUEDEN UNIR BAJO CIERTAS CIRCUNSTANCIAS FORMANDO UNA NUEVA RELACIÓN

SE PUEDEN UNIR DOS O MÁS RELACIONES.

R	(supplier	part)	S	(part	project)
1	1	1	1	1	
2	1	1	1	2	
2	2	2	2	1	

FIG. 5. Two joinable relations

R*S	(supplier	part	project)
	1	1	1
	1	1	2
	2	1	1
	2	1	2
	2	2	1

FIG. 6. The natural join of R with S (from Figure 5)

U	(supplier	part	project)
	1	1	2
	2	1	1
	2	2	1

FIG. 7. Another join of R with S (from Figure 5)



# JOIN

EN CASO DE EXISTIR AMBIGÜEDAD EN SE PUEDE ACUDIR A PROPIEDADES MATEMÁTICAS BASADAS EN LA TEORÍA DE LOS CONJUNTOS.

$$(1) \pi_1(T) = \pi_2(S),$$

$$(2) \pi_2(T) = \pi_1(R),$$

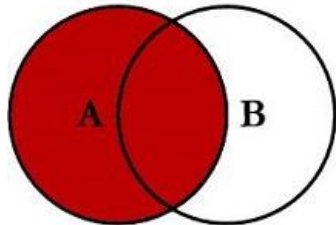
$$(3) T(j, s) \rightarrow \exists p (R(S, p) \wedge S(p, j)),$$

$$(4) R(s, p) \rightarrow \exists j (S(p, j) \wedge T(j, s)),$$

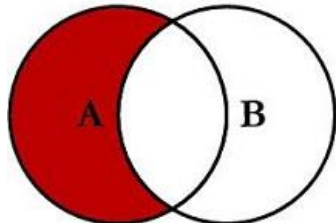
$$(5) S(p, j) \rightarrow \exists s (T(j, s) \wedge R(s, p)),$$



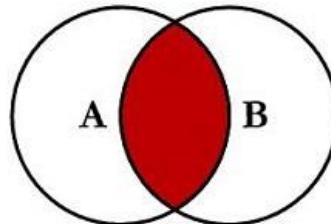
# SQL JOINS



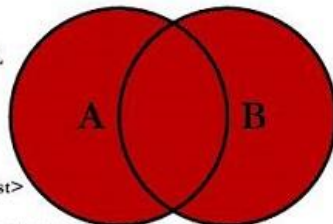
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



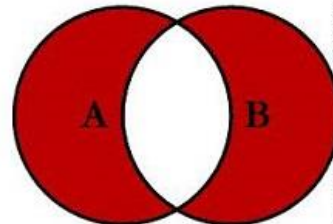
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```



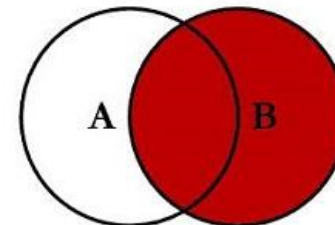
```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



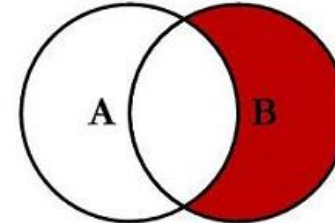
```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```



```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```

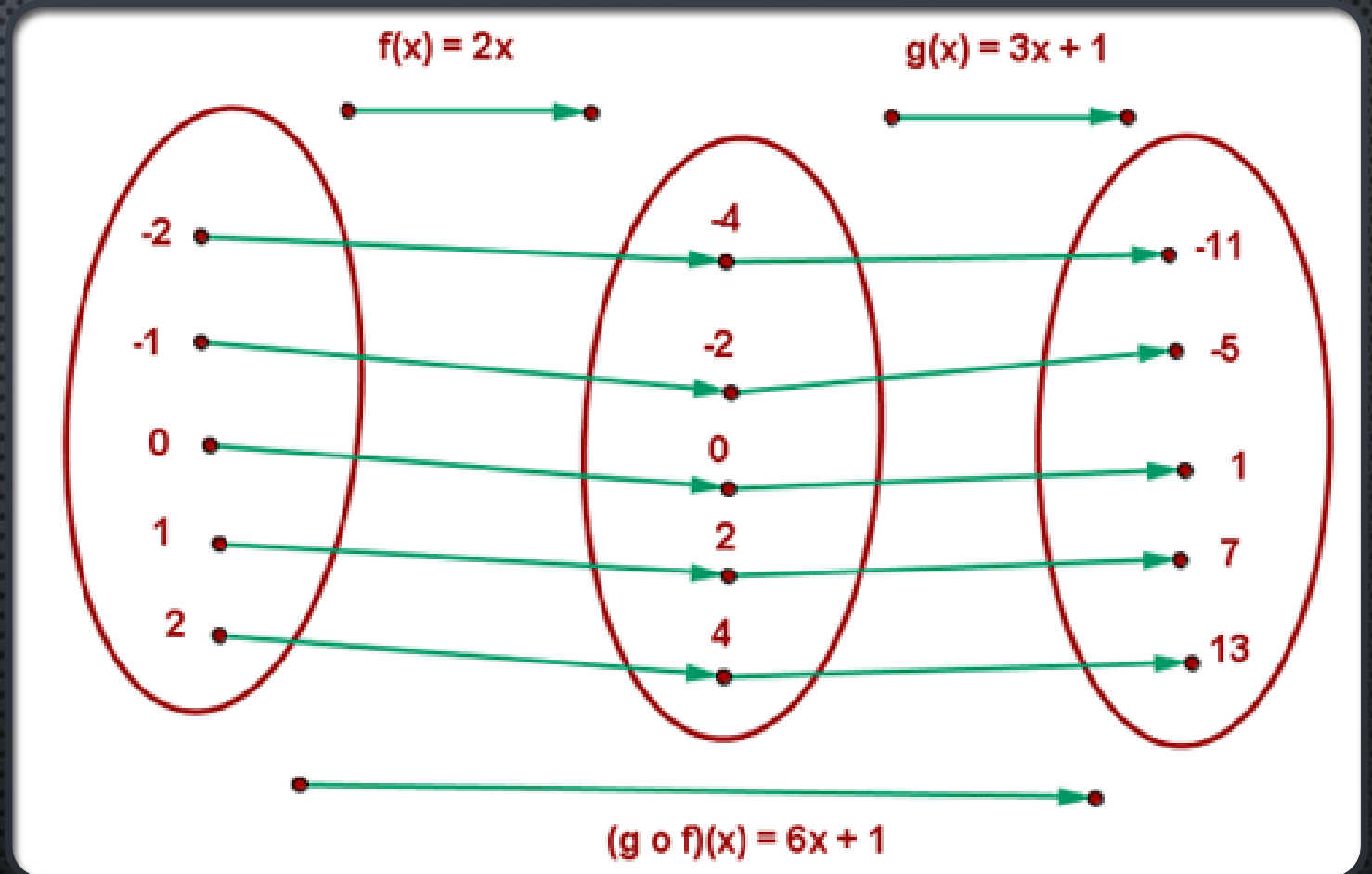


```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```

# COMPOSICIÓN

DOS RELACIONES SE PUEDEN COMPONER SI Y SOLO SI SE PUEDEN UNIR. SIN EMBARGO, LA EXISTENCIA DE MÁS DE UNA UNIÓN DE  $R$  CON  $S$  NO IMPLICA LA EXISTENCIA DE MÁS DE UNA COMPOSICIÓN DE  $R$  CON  $S$ .

$$R \cdot S = \pi_{13}(R * S).$$



# RESTRICCIÓN.

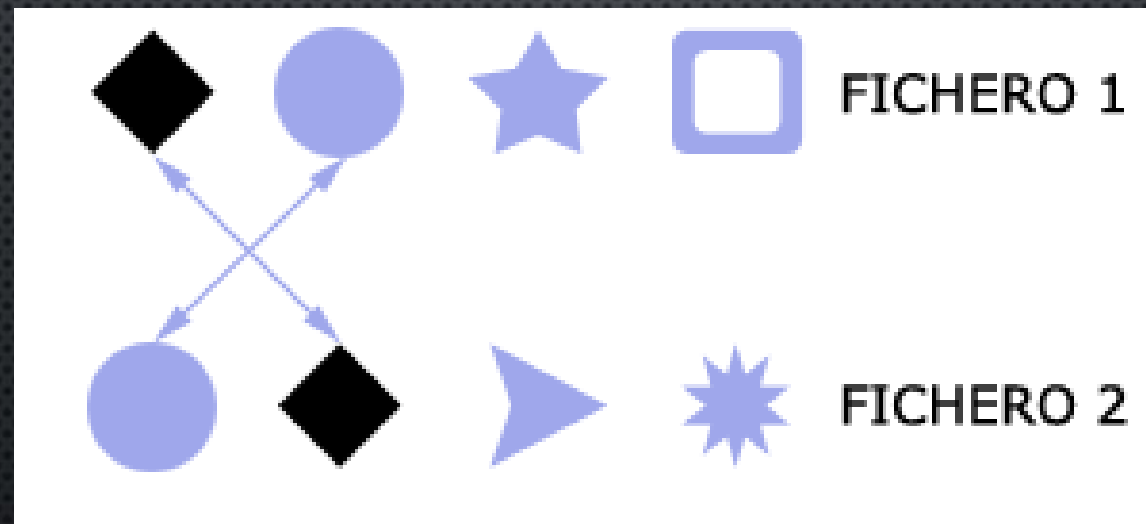
LA RESTRICCIÓN ES UNA FORMA EN LA QUE PODEMOS USAR UNA RELACIÓN S PARA RESTRINGIR UNA RELACIÓN Y LLEGAR A UN SUBCONJUNTO MÁS FÁCIL.

R	(s	p	j)	S	(p	j)	R'	(s	p	j)
	1	a	A		a	A		1	a	A
	2	a	A		c	B		2	a	A
	2	a	B		b	B		2	b	B
	2	b	A							
	2	b	B							

FIG. 13. Example of restriction



# REDUNDANCIA





# REDUNDANCIA FUERTE

SI SE TIENE UNA RELACIÓN QUE TIENE UNA PROYECCIÓN QUE SE DERIVA DE OTRA PROYECCIÓN DE OTRAS RELACIONES.



# REDUNDANCIA DÉBIL.

CUANDO UN CONJUNTO DE RELACIONES CONTIENE UNA RELACIÓN QUE NO SE DERIVA DE OTROS MIEMBROS SINO QUE LLEGA DE LA UNIÓN DE RELACIONES, DICHO DE OTRO MODO SI SE TIENE UNA TABLA QUE SE DERIVE POR OPERACIONES CONJUNTAS DE OTRAS TABLAS EN ESE CONJUNTO DE RELACIONES ESA TABLA ES DÉBILMENTE REDUNDANTE



# CONSISTENCIA

ELIMINAR O CONTROLAR LAS  
REDUNDANCIAS QUE EXISTEN

# MANEJO DE BASE DE DATOS EN LA ACTUALIDAD

EL MANEJO DE BASE DE DATOS ACTUALES EN SU MAYORÍA SON RELACIONALES Y COMO SE PUDO VER A LO LARGO DE LA PRESENTACIÓN GRAN PARTE DE LOS CONCEPTOS ACTUALES EN BASES DE DATOS RELACIONALES SE DEBEN A LAS APORTACIONES DE CODD.

GRACIAS