

# Curso de Base de Datos

2022-2

**Semana 3**  
**Modelo Entidad Relación**  
**IDEF1X**  
**Barker**



Profesor del curso:  
César Aguilera  
Luis Ríos



Elaborado por:  
César Aguilera  
Luis Ríos



Revisado por:  
Rony Cueva  
César Aguilera

# Saberes previos

- El modelo conceptual: análisis formal para representar la realidad con un modelo semántico
- Herramienta a utilizar: Diagrama Entidad Relación (DER)
- Notación en el DER: entidad, atributos y relación

- Colección de herramientas conceptuales para describir datos, relaciones de datos, semántica de datos y restricciones de datos.
- Tipos:
  - *Object-based logical models*
    - Modelo Entidad Relación
  - *Record-based logical models*
    - Modelo Relacional
  - *Physical data models*

## Elaboración del Modelo Entidad Relación

- Identificar entidades
- Definir atributos
- Establecer relaciones
- Precisar cardinalidades

- El Modelo Relacional:
  - Estructura y operaciones relacionales
  - Reglas de integridad
- Técnicas de modelamiento de datos
  - IDEF1X
    - Tipos de datos
    - Modelamiento y convenciones
  - Notación Barker

## Descripción

Los datos se agrupan como relaciones de datos y se representan en forma de tablas, donde cada fila corresponde a una "tupla" de la relación y cada columna representa a un dominio. No existe otra forma de agrupar datos. Cada dato es atómico, representa un solo valor del dominio.

Grado de la relación : Cantidad de columnas de la tabla

Cardinalidad : Cantidad de filas

*(1ra regla de Codd - 1985 “Regla de la información” : Toda información debe expresarse sólo en términos de valores en relaciones).*

Cada columna de la relación representa diferente información. Aunque dos columnas de una misma relación pueden tener exactamente el mismo dominio, representarán información diferente. En efecto, cada columna es un uso particular de un dominio.

El modelo relacional es un modelo de datos y, como tal, tiene en cuenta los tres aspectos siguientes de los datos:

1. La estructura, que debe permitir representar la información que nos interesa del mundo real.
2. La manipulación, a la que da apoyo mediante las operaciones de actualización y consulta de los datos.
3. La integridad, que es facilitada mediante el establecimiento de reglas de integridad; es decir, condiciones que los datos deben cumplir



Además:

- Representación de los datos como una colección de tablas. Cada tabla tiene un nombre único. Contiene un solo tipo de registro.
- Las filas (tuplas) representan relaciones entre un conjunto de valores. Tiene un número fijo de atributos. No existen filas duplicadas.
- Las columnas son los dominios de los atributos.

# Lenguajes Formales (teóricos)

- Algebra Relacional
  - Tipo Procedural
- Cálculo Relacional
  - Tipo Non Procedural

- Fundamentales
  - SELECT (  $\sigma$  )
  - PROJECT (  $\pi$  )
  - CARTESIAN PRODUCT (  $\times$  )
  - UNION (  $\cup$  )
  - SET DIFFERENCE (  $-$  )
- Adicionales
  - Set Intersection (  $\cap$  )
  - Theta Join (  $|x|_{\theta}$  )
  - Natural Join (  $|x|$  )
  - Division (  $\div$  )

## Estructura

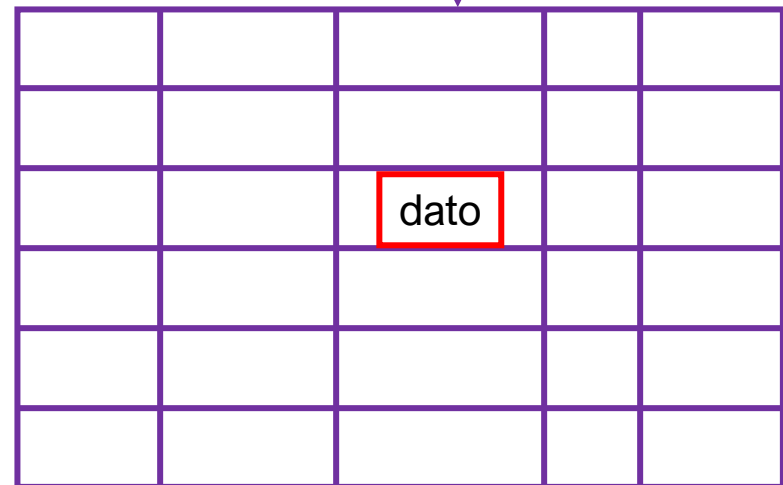
**Dato** :Unidad básica (atómica), representa un solo valor del dominio.

**Relación** (de datos) : Estructura básica (compuesta)

**Tabla** : Representación de una relación entre dominios de datos

**Columna** : Representa a un dominio

**Fila** : representa a una tupla de la relación



		dato		

## Estructura : restricciones

### Tabla

No tiene conceptos posicionales: no interesa el orden de las filas, ni el de las columnas.

### Filas

Contenido único respecto a la Relación: no se permiten filas duplicadas en una tabla.

### Llave primaria (PK)

Propiedad de unicidad.- El valor de la llave primaria en cada fila identifica a esa fila de manera única (distingue a la fila de cualquier otra de la tabla).

Propiedad de mínima cantidad de columnas .- Si la llave primaria es compuesta y una de sus columnas es eliminada, la propiedad de unicidad ya no se cumple.

## Estructura

### Llave primaria (PK)

Propiedad de unicidad interpretada como propiedad de identificación .- El valor de la llave primaria en cada fila identifica al objeto particular representado por esa fila dentro de la clase de objetos que representa esa relación. Si en el modelo existe la necesidad de referirse al mismo objeto, este mismo valor es usado.

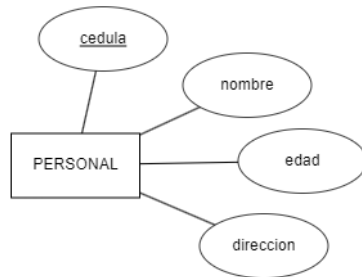
### Llave foránea (FK)

Columna de una tabla que contiene los valores de un dominio que sirven al mismo de la llave primaria en otra(s) tabla(s) para identificar al mismo objeto.

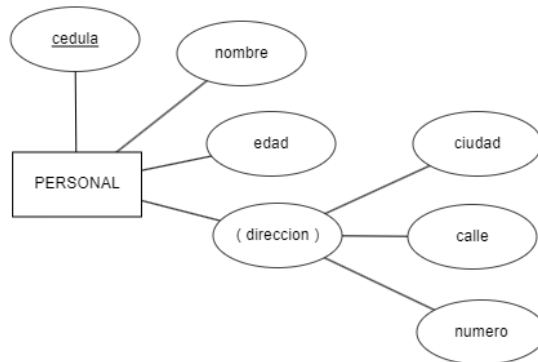
## Paso de DER a Relacional(Tablas)

- **Pocas “reglas”:**
  - Por entidad del esquema conceptual se crea una tabla
    - Se selecciona uno de los atributos determinantes de la entidad como clave primaria de la tabla
  - Las relaciones n:n se modelan como una tabla nueva, donde se colocan las claves primarias de las tablas que representan a cada una de las entidades participantes
    - Si existen atributos en la relación se tratan como si fueran los de una entidad
  - Por cada entidad débil (relaciones 1:n) se crea una tabla. Se incluye como atributos los de la clave primaria de la tabla que representa a la entidad fuerte
    - La clave primaria la conforman la clave primaria de la tabla que representa a la entidad fuerte + atributo(s) que representan al identificador parcial

## Paso de DER a Relacional(Tablas)



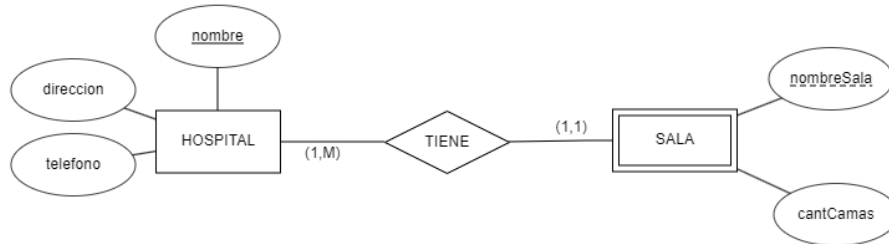
PERSONAL(cedula, nombre, edad, direccion)



PERSONAL(cedula, nombre, edad, ciudad, calle, numero)

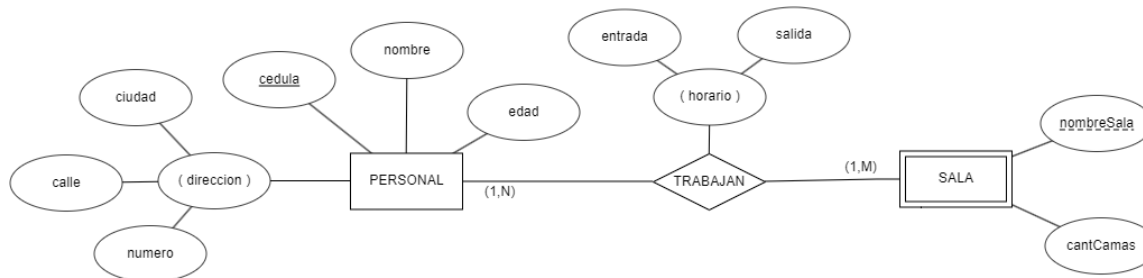


## Paso de DER a Relacional(Tablas)



HOSPITAL(nombre, dirección, teléfono)

SALA(nombreHospital, nombreSala, cantCamas)



HOSPITAL(nombre, dirección, teléfono)

SALA(nombreHospital, nombreSala, cantCamas)

TRABAJA(nombreSala, cedula, entrada, salida)

## Operaciones Relacionales

El álgebra relacional es un sistema cerrado de operaciones definidas sobre relaciones.

Los operandos y los resultados son relaciones.

Pueden combinarse (los resultados de una pueden ser operandos de otras).

**Fundamentales :** Selección  
Proyección  
Producto Cartesiano  
Unión  
Diferencia

**Derivadas :** Intersección  
Conjunción natural (Equi-Join)

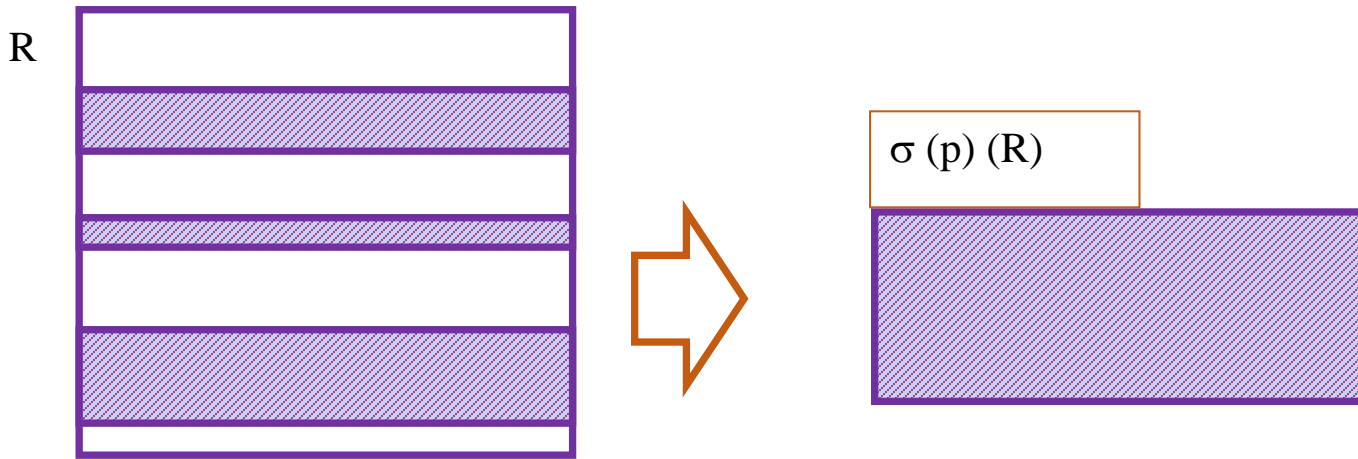
**De modificación :** Asignación / Inserción / Eliminación / Actualización

# Operación Relacional-Fundamentales

## Selección

$$\sigma(p)(R)$$

- Operación unitaria que produce una relación con tuplas seleccionadas de la relación operando (R) que satisfagan un predicado (p).
- En el predicado :
  - Los operandos pueden ser columnas o constantes
  - Los operadores pueden ser de comparación (=,  $\neq$ , < , <=, > , >= ) aritméticos (+, -, \*, /, \*\*) y lógicos (Y, O,  $\neg$ )



# Operación Relacional-Fundamentales

Donde  $r$  es la relación sobre la que se actúa y  $P$  es el predicado que debe cumplirse

**Empleado**

Número de Empleado	Nombre del Empleado	Salario	Código Depto.
1	Pedro	12,000	VE
2	Luís		NO
3	Francisco	36,000	
4	Lorena	24,000	
5	Gabriela	24,000	NO

La Selección de todos los empleados del departamento de nómina, es:

$\sigma_{\text{Cod.Depto}='NO'}(\text{Empleado})$

**RESULTADO**

Número de Empleado	Nombre del Empleado	Salario	Código Depto.
2	Luís		NO
5	Gabriela	24,000	NO

# Operación Relacional-Fundamentales

## Proyección

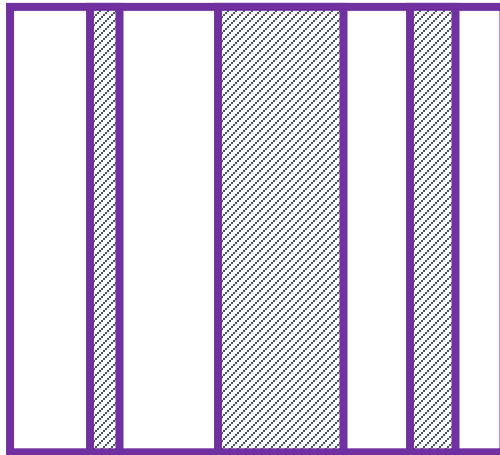
$\pi (A) (R)$

Operación unitaria que produce una relación con las columnas de la relación operando (R) que se den como argumento (A) y eliminando de este resultado las tuplas repetidas.

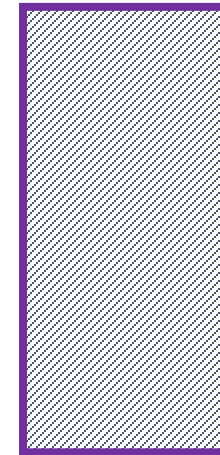
El argumento :

Lista de columnas  $A = (a1, a2, ..., an)$

R



$\pi (A) (R)$



# Operación Relacional-Fundamentales

Sintaxis:

$\Pi_{A1, \dots, An}(r)$

Donde  $A1, \dots, An$  es la lista de atributos y "r" la relación sobre la que se actúa.

Empleado			
Número de Empleado	Nombre del Empleado	Salario	Código Depto.
1	Pedro	12,000	VE
2	Luís		NO
3	Francisco	36,000	
4	Lorena	24,000	
5	Gabriela	24,000	NO

La Proyección de los atributos Nombre y Número de empleado, es:

$\Pi_{\text{Nombre del Empleado, Número de empleado}}(\text{Empleado})$

## RESULTADO

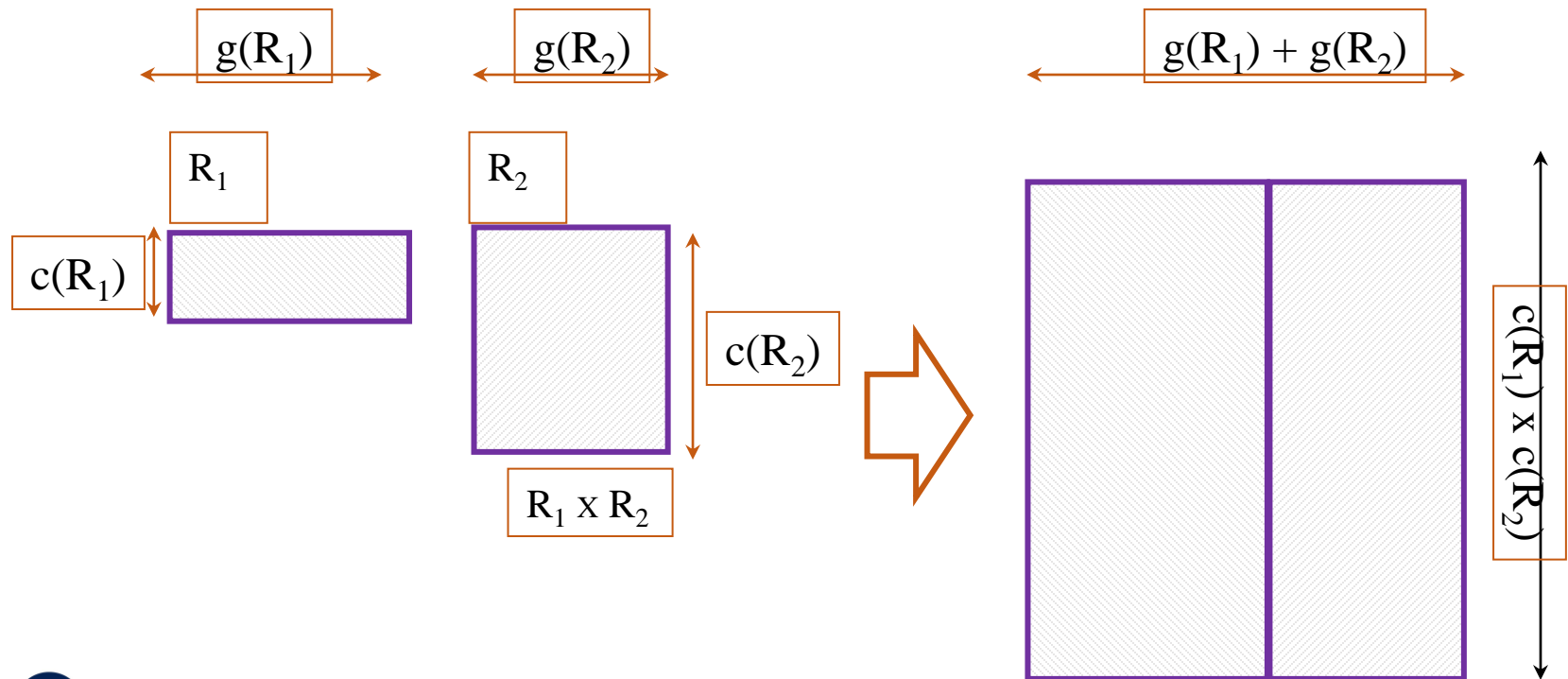
Nombre del Empleado	Número de Empleado
Pedro	1
Luís	2
Francisco	3
Lorena	4
Gabriela	5

# Operación Relacional-Fundamentales

## Producto Cartesiano

$$R_1 \times R_2$$

Operación binaria que produce una relación que contiene como tuplas a todas las combinaciones de las tuplas de la primera relación operando ( $R_1$ ) con las de la segunda ( $R_2$ ). La tabla resultado tiene como columnas la concatenación de las columnas de ambas tablas operandos.



# Operación Relacional-Fundamentales

Sintaxis:  $(r_1 \times r_2)$

Donde  $r_1$  y  $r_2$  son las relaciones sobre las que se actúa

**Departamento**

Cod. Depto.	Nombre del Depto.
VE	Ventas
NO	Nómina
IN	Investigación
ME	Mercadotecnia
RE	Resultados

**Empleado**

Número de Empleado	Nombre del Empleado	Salario	Cod. Depto.
1	Pedro	12,000	VE
2	Luís		NO
3	Francisco	36,000	
4	Lorena	24,000	
5	Gabriela	24,000	NO

El PRODUCTO CARTESIANO de las relaciones anteriores, es:

(Departamento X Empleado)

**RESULTADO**

Cod. Depto.	Nombre del Depto.	Número de Empleado	Nombre de Empleado	Salario	Cod. Depto.
VE	Ventas	1	Pedro	12,000	VE
VE	Ventas	2	Luís		NO
VE	Ventas	3	Francisco	36,000	
VE	Ventas	4	Lorena	24,000	
VE	Ventas	5	Gabriela	24,000	NO
NO	Nómina	1	Pedro	12,000	VE
NO	Nómina	2	Luís		NO
NO	Nómina	3	Francisco	36,000	
...	...				
RE	Resultados	5	Gabriela	24,000	NO



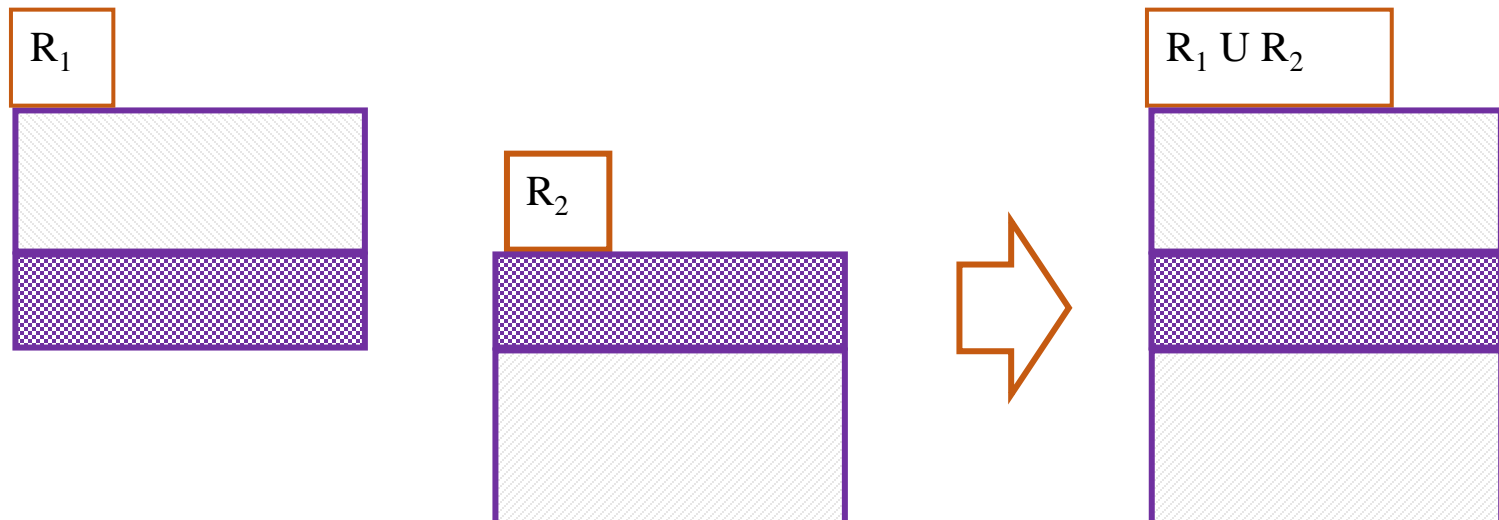
# Operación Relacional-Fundamentales

## Unión

$$R_1 \cup R_2$$

Operación binaria que produce una relación que contiene todas las tuplas de la primera relación operando ( $R_1$ ) y todas las de la segunda ( $R_2$ ) incluyendo sólo una vez aquellas comunes (evitando repeticiones).

Ambas relaciones operandos deben ser “compatibles para unirse” : deben ser del mismo grado y los dominios de las columnas (por lo menos de una) de ambas relaciones deben ser iguales.



# Operación Relacional-Fundamentales

Sintaxis:

$r_1 \cup r_2$

Donde  $r_1$  y  $r_2$  son las relaciones sobre las que se actúa.

**Empleados Antiguos**

Número de Empleado	Nombre del Empleado	Salario
1	Pedro	12,000
2	Luís	
3	Francisco	36,000

**Empleados Nuevos**

Número de Empleado	Nombre del Empleado	Salario
3	Francisco	36,000
4	Lorena	24,000
5	Gabriela	24,000

La UNIÓN de las relaciones anteriores, es:

**Empleados Antiguos U Empleados Nuevos**

**RESULTADO**

Número de Empleado	Nombre del Empleado	Salario
1	Pedro	12,000
2	Luís	
3	Francisco	36,000
4	Lorena	24,000
5	Gabriela	24,000

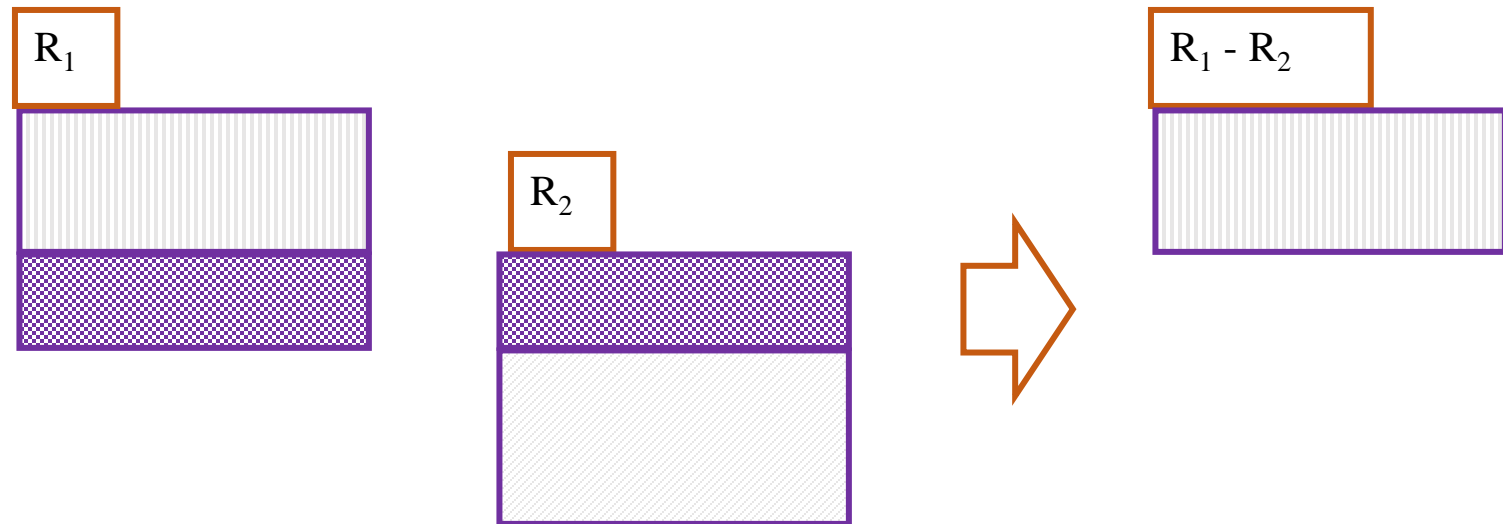
# Operación Relacional-Fundamentales

## Diferencia

$$R_1 - R_2$$

Operación binaria que produce una relación que contiene todas las tuplas de la primera relación operando ( $R_1$ ) que no se encuentren en la segunda ( $R_2$ ).

Ambas relaciones operandos deben ser “compatibles para unirse”.



# Operación Relacional-Fundamentales

## Ejemplo:

La diferencia de dos relaciones R y S con el mismo esquema.

NIF	Nombre	Localidad
1254	Juan	Lima
1578	Sergio	Trujillo
1865	Patricia	Arequipa
1904	Fernanda	Cuzco

R

R - S

NIF	Nombre	Localidad
1254	Juan	Lima
1904	Fernanda	Cuzco

S

NIF	Nombre	Localidad
1248	Nancy	Tumbes
1578	Sergio	Trujillo
1865	Patricia	Arequipa
1908	Andrés	Iquitos

# Operación Relacional-Derivadas

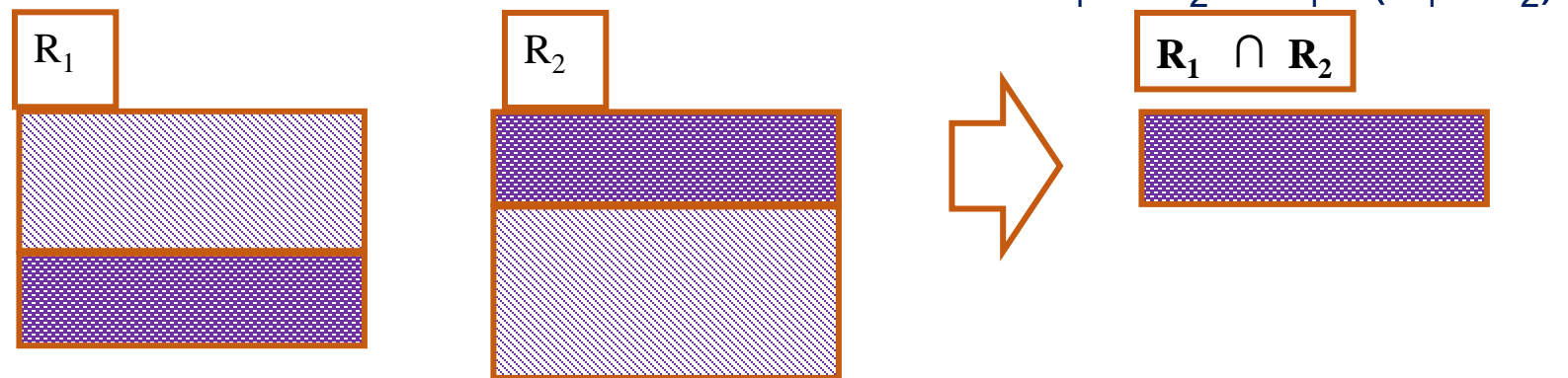
## Intersección

$$R_1 \cap R_2$$

Operación binaria que produce una relación que contiene todas las tuplas de la primera relación operando ( $R_1$ ) que también están en la segunda relación operando ( $R_2$ ).

Ambas relaciones operandos deben ser “compatibles para unirse”.

Es una operación derivada pues puede reemplazarse por un par de diferencias :



# Operación Relacional-Fundamentales

Sintaxis:  $(r_1 \cap r_2)$  ó  $r_1 - (r_1 - r_2)$

Donde  $r_1$  y  $r_2$  son las relaciones sobre las que se actúa

**Empleados Antiguos**

Número de Empleado	Nombre del Empleado	Salario
1	Pedro	12,000
2	Luís	
3	Francisco	36,000

**Empleados Nuevos**

Número de Empleado	Nombre del Empleado	Salario
3	Francisco	36,000
4	Lorena	24,000
5	Gabriela	24,000

La INTERSECCIÓN de las relaciones anteriores, es:

**Empleados Antiguos  $\cap$  Empleados Nuevos**

**RESULTADO**

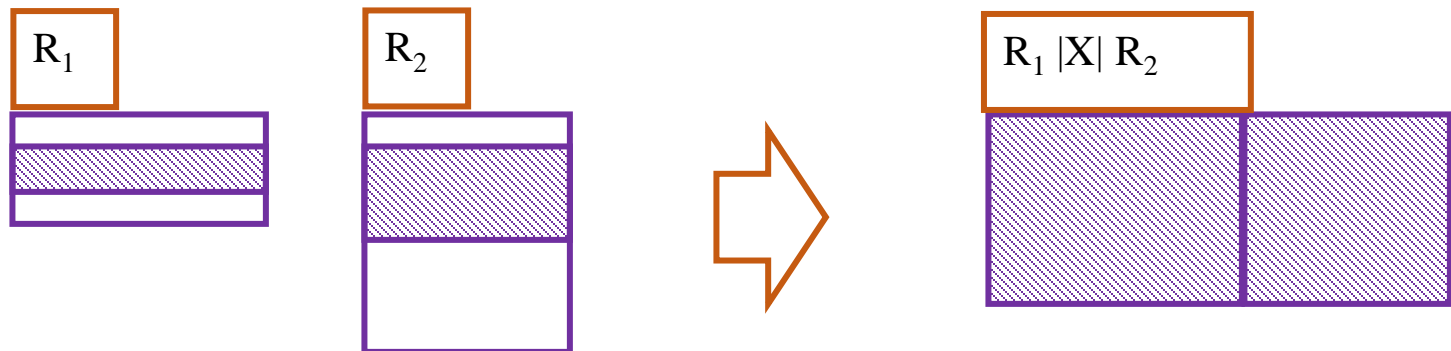
Número de Empleado	Nombre del Empleado	Salario
3	Francisco	36,000

# Operación Relacional-Derivadas

## Conjunción Natural (Equi-Join)

$R_1 \bowtie R_2$

Caso particular de conjunción donde el predicado (condición de conjunción) es de comparación por igualdad de cada pareja de columnas de las tablas operando ( $R_1$  y  $R_2$ ) que tengan igual dominio y significado (no necesariamente igual nombre de columna).



# Operación Relacional-Fundamentales

Sintaxis:  $(r_1 \theta_P r_2)$  ó  $\sigma_P (r_1 \times r_2)$

Donde  $r_1$  y  $r_2$  son las relaciones sobre las que se actúa y  $P$  representa el predicado de la selección.

**Departamento**

Cod. Depto.	Nombre del Depto.
VE	Ventas
NO	Nómina
IN	Investigación
ME	Mercadotecnia
RE	Resultados

**Empleado**

Número de Empleado	Nombre del Empleado	Salario	Cod. Depto.
1	Pedro	12,000	VE
2	Luís		NO
3	Francisco	36,000	
4	Lorena	24,000	
5	Gabriela	24,000	NO

El JOIN con la columna en común del código del departamento en las relaciones anteriores, es:

(Departamento  $\theta_{\text{Departamento.codDepto} = \text{Empleado.codDepto}}$  Empleado)

**RESULTADO**

Cod. Depto.	Nombre del Depto.	Número de Empleado	Nombre de Empleado	Salario
VE	Ventas	1	Pedro	12,000
NO	Nómina	2	Luís	
NO	Nómina	5	Gabriela	24,000

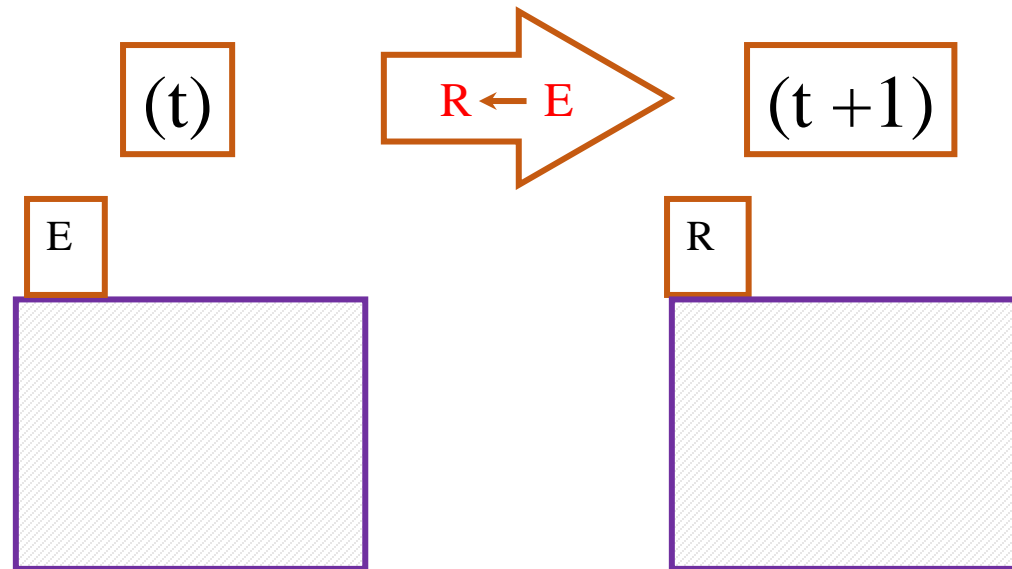


# Operación Relacional-Modificación

## Asignación

$$R \leftarrow E$$

Operación que asigna a una relación que el usuario pone nombre (R) el resultado de una expresión relacional (E).

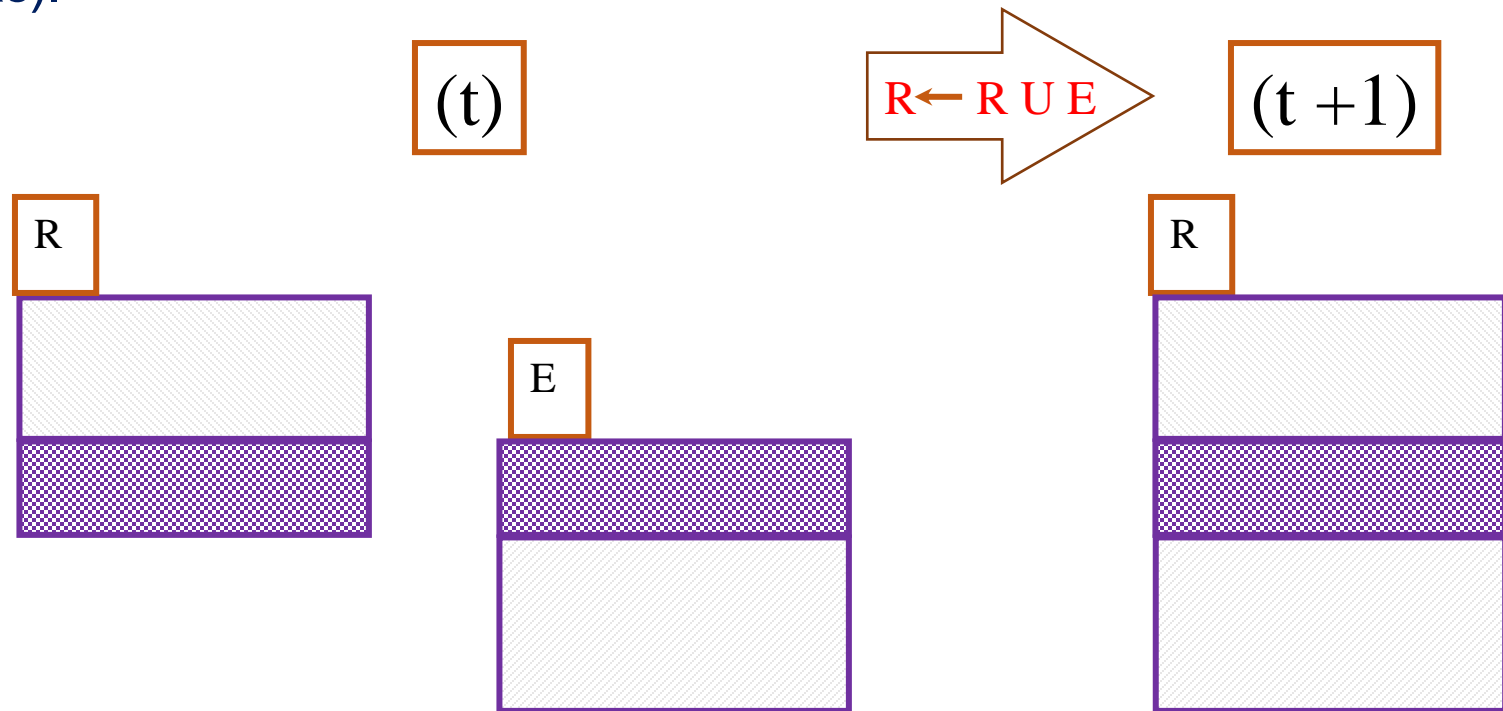


# Operación Relacional-Modificación

## Inserción

$$R \leftarrow R \cup E$$

Operación que permite insertar una relación de tuplas, que puede ser resultado de una expresión relacional (E), a una relación (R), excluyendo las que ya se encuentren en R (evitando repetición de filas).

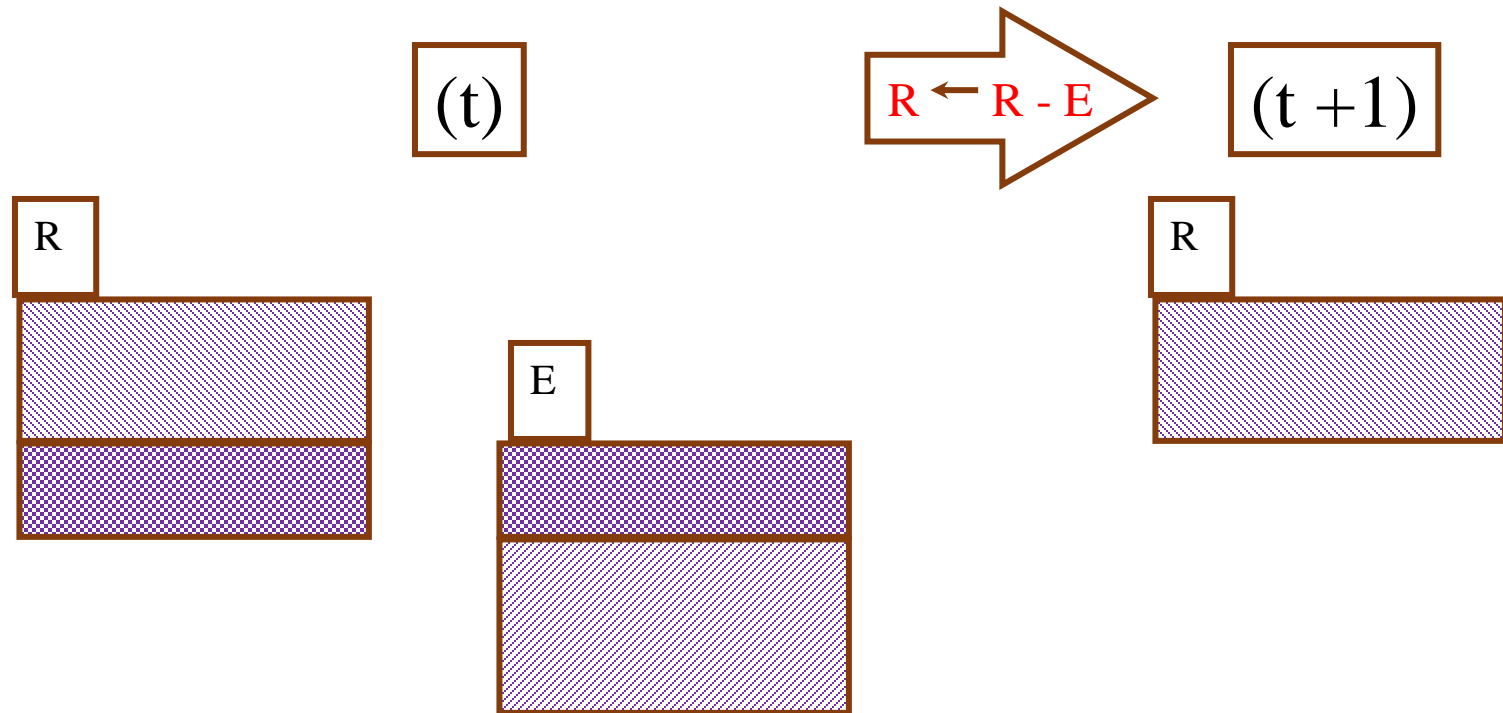


# Operación Relacional-Modificación

## Eliminación

$$R \leftarrow R - E$$

Operación que permite eliminar una relación de tuplas, que puede ser resultado de una expresión relacional (E), de una relación (R).

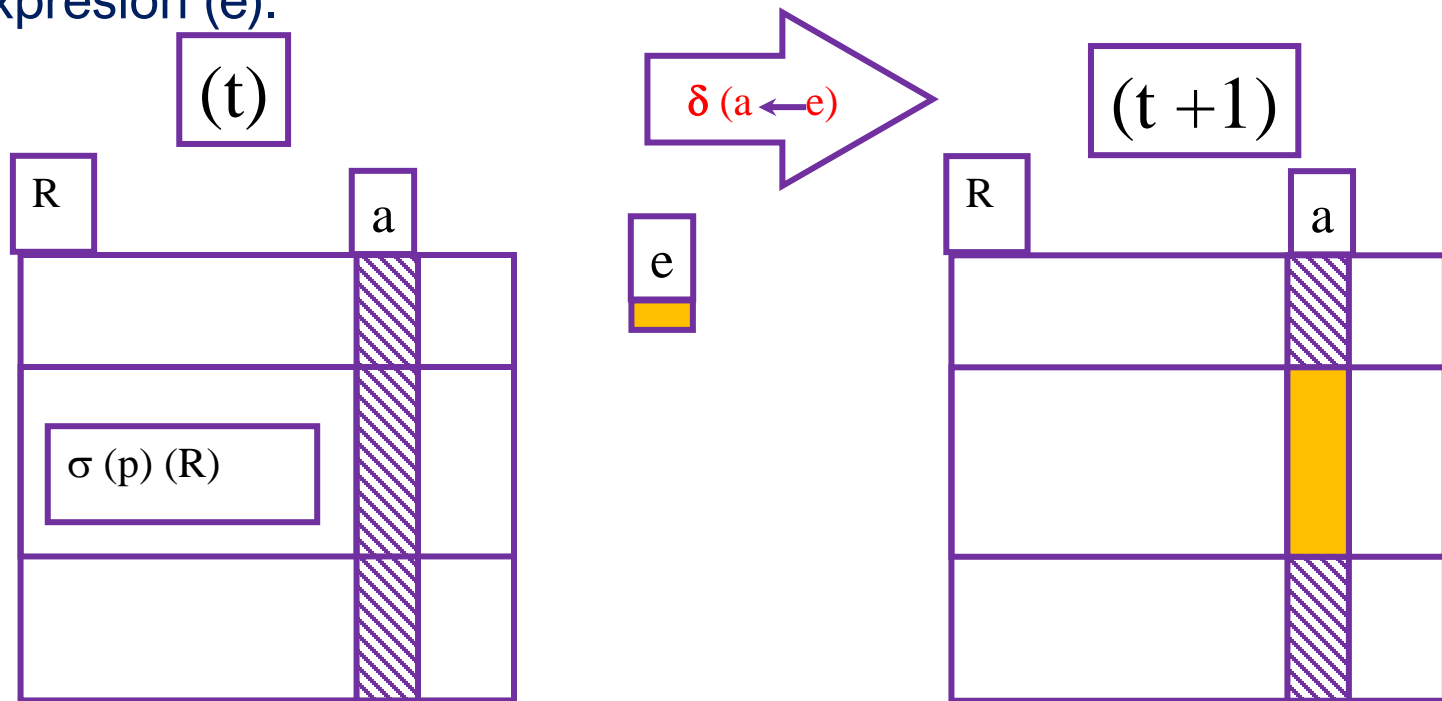


# Operación Relacional-Modificación

## Actualización

$$\delta(a \leftarrow e)(\sigma(p)(R))$$

Operación que permite cambiar el valor de un atributo (a) a las tuplas seleccionadas de la relación operando (R) que satisfagan un predicado (p), por otro valor que a su vez puede ser resultado de una expresión (e).



# Ejemplo:

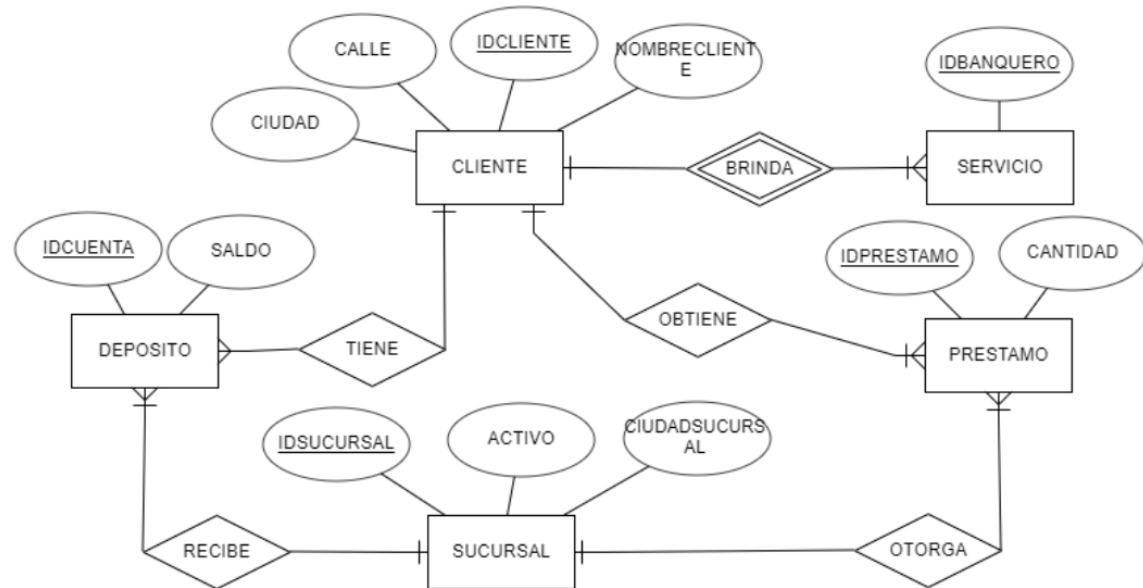
Depósito **D** (IdSucursal, IdCuenta, IdCliente, Saldo)

Cliente **C** (IdCliente, NCliente, Calle, CiudadCliente)

Préstamo **P** (IdSucursal, IdPrestamo, IdCliente, Cantidad)

Sucursal **S** (IdSucursal, Activo, CiudadSucursal)

Servicio **R** (IdCliente, IdBanquero)



# Ejemplo:

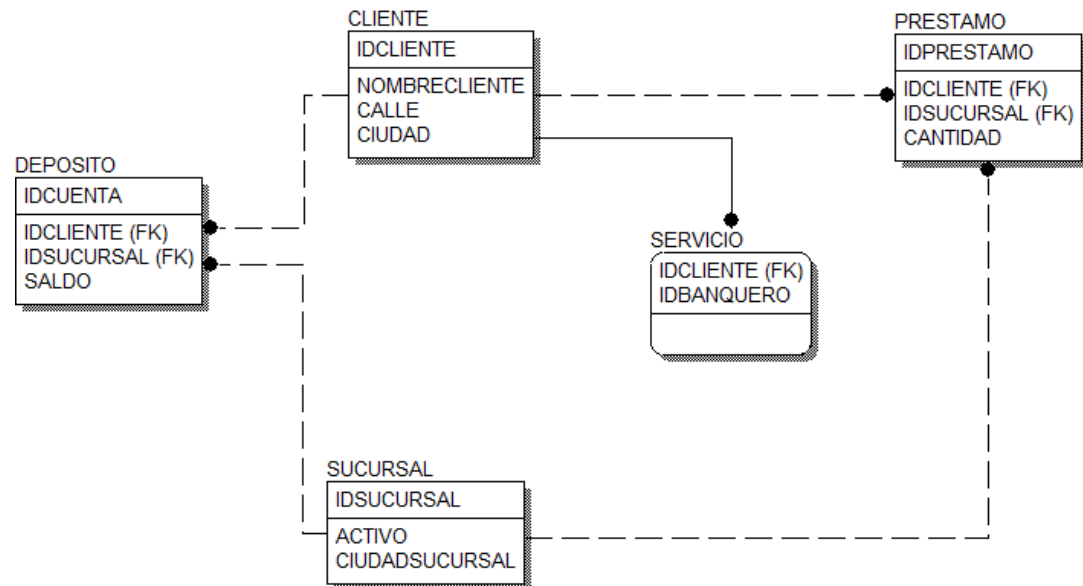
Depósito **D** (IdSucursal, IdCuenta, IdCliente, Saldo)

Cliente **C** (IdCliente, NCliente, Calle, CiudadCliente)

Préstamo **P** (IdSucursal, IdPrestamo, IdCliente, Cantidad)

Sucursal **S** (IdSucursal, Activo, CiudadSucursal)

Servicio **R** (IdCliente, IdBanquero)



## Ejemplo:

### Tabla Préstamo

IDSUCURSAL	IDPRESTAMO	IDCLIENTE	CANTIDAD
S1	001	C1	500
S1	002	C2	900
S2	003	C3	1100
S3	004	C4	5000
S2	005	C5	200
S1	006	C6	2000

Se pide: los préstamos de la sucursal S1 mayores a 1000

## Ejemplo:

### Selección $\sigma$ (p) (R)

P: predicado  $\rightarrow$  “ cantidad >1000 Y idsucursal='S1' ”

R: relación  $\rightarrow$  Préstamos

$\sigma$  (cantidad > 1000 y idsucursal='S1') (Préstamos)

En SQL

SELECT columnas seleccionadas

FROM de que tablas

WHERE condición que deben cumplir las filas seleccionadas

SELECT \*

FROM Prestamos

WHERE cantidad > 1000 AND idsucursal='S1'



## Ejemplo:

### Proyección

$$\pi (A) (R)$$

A: relación de columnas

R: relación  $\rightarrow$  Cliente

Se pide: las ciudades de la tabla cliente

IDCLIENTE	NOMBRECLI	CALLE	CIUDAD
C1	.....	.....	LIMA
C2			TRUJILLO
C3			LIMA
C4			CUSCO
C5			LIMA

**IMPORTANTE:** Se eliminan las tuplas repetidas

# Ejemplo:

## Proyección

$$\pi (A) (R)$$

$$\pi (\text{Ciudad}) (R)$$

R: relación  $\rightarrow$  Cliente

$$\pi (\text{Ciudad}) (\text{Cliente})$$

En SQL:

SELECT ciudad FROM Cliente

CIUDAD
LIMA
TRUJILLO
LIMA
CUSCO
LIMA

IMPORTANTE: Se eliminan las tuplas repetidas

En SQL:

SELECT **DISTINCT** ciudad  
FROM Cliente

CIUDAD
LIMA
TRUJILLO
CUSCO

## Ejemplo:

### Producto Cartesiano $R_1 \times R_2$

R1: Cliente; R2: Servicio

$R_1 \times R_2$

IDCLIENTE	NOMBRECLI	CALLE	CIUDAD
C1	.....	.....	LIMA
C2			TRUJILLO
C3			LIMA
C4			CUSCO
C5			LIMA

R1: Cliente

IDCLIENTE	IDBANQUERO
C1	B1
C2	B2
C3	B3
C4	B4
C5	B2
C1	B3
C3	B2

R2: Servicio

## Ejemplo:

### Producto Cartesiano $R_1 \times R_2$

R1: Cliente; R2: Servicio.  $R_1 \times R_2$

IDCLIENTE	NOMBRECLI	CALLE	CIUDAD	IDCLIENTE	IDBANQUERO
C1	ANDRES	AV.BOLIVAR	LIMA	C1	B1
...	...	...	...	C2	B2
...	...			C3	B3
				C4	B4
				C5	B2
				C1	B3
				C3	B2
C2	CECILIA	AV. 2 DE MAYO	TRUJILLO	C1	B1
...	...	...	...	C2	B2
...				C3	B3

## Ejemplo:

### Producto Cartesiano $R_1 \times R_2$

R1: Cliente; R2: Servicio .  $R_1 \times R_2$

Se pide: todos los datos de todos los clientes atendidos por el banquero “B3”

IDCLIENTE	NOMBRECLI	CALLE	CIUDAD	IDCLIENTE	IDBANQUERO
C1	ANDRES	AV.BOLIVAR	LIMA	C1	B3
C3	PEDRO	DEL SOLAR	LIMA	C3	B3

$\sigma (R2.Idbanquero='B3' \text{ Y } R1.idcliente=R2.idcliente) (R1 \times R2)$

En SQL:

SELECT \*

FROM Cliente C, Servicio R

WHERE C.Idcliente=R.Idcliente AND R.Idbanquero='B3'

## Ejemplo:

### Producto Cartesiano $R_1 \times R_2$

R1: Cliente; R2: Servicio.  $R_1 \times R_2$

Se pide: el código y el nombre de todos los clientes atendidos por el banquero “B3”.

$\pi (R1.idcliente, R1.nombrecli) \sigma (R2.idbanquero='B3' \text{ Y } R1.idcliente=R2.idcliente) (R1 \times R2)$

En SQL:

```
SELECT C.idcliente, C.Nombre  
FROM Cliente C, Servicio R  
WHERE C.idcliente=R.idcliente AND R.idbanquero='B3'
```

## Ejemplo:

### Unión $R_1 \cup R_2$

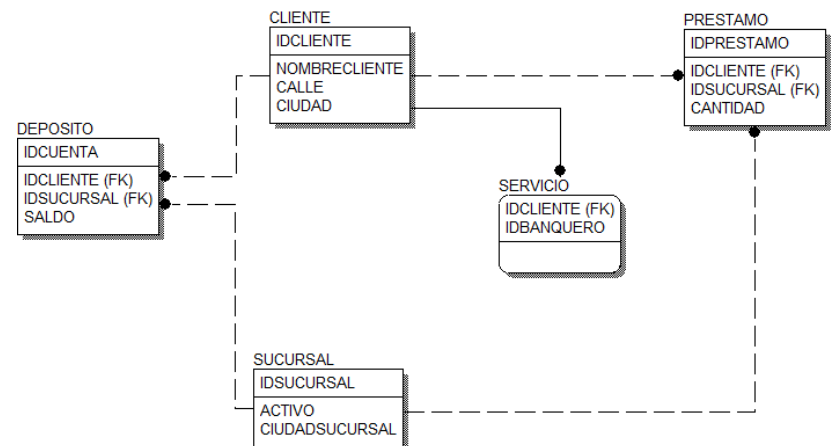
Ejemplo: Los clientes que tengan depósitos o préstamos en la sucursal S5

$V1 \leftarrow \pi (\text{Idcliente}) (\sigma(\text{Idsucursal}='S5')(D))$  [Depósitos]

$V2 \leftarrow \pi (\text{Idcliente}) (\sigma(\text{Idsucursal}='S5')(P))$  [Préstamos]

Ahora para obtener a los que tienen depósitos o préstamos

$V3 \leftarrow V1 \cup V2$



## Ejemplo:

### Unión $R_1 \cup R_2$

Ejemplo: Los clientes que tengan depósitos o préstamos en la sucursal S5

$V1 \leftarrow \pi (\text{Idcliente}) (\sigma(\text{Idsucursal}='S5')(D))$  [Depósitos]

$V2 \leftarrow \pi (\text{Idcliente}) (\sigma(\text{Idsucursal}='S5')(P))$  [Préstamos]

Ahora para obtener a los que tienen depósitos o préstamos

$V3 \leftarrow V1 \cup V2$

En SQL:

```
SELECT Idcliente  
FROM Deposito  
WHERE Idsucursal ='S5'
```

**UNION**

```
SELECT Idcliente  
FROM Prestamo  
WHERE Idsucursal ='S5'
```



## Ejemplo:

### Diferencia $R_1 - R_2$

Ejemplo: Los clientes de S5 que tienen depósito pero no préstamos

$V1 \leftarrow \pi (\text{Idcliente}) (\sigma(\text{Idsucursal}='S5')(D))$  [Depósitos]

$V2 \leftarrow \pi (\text{Idcliente}) (\sigma(\text{Idsucursal}='S5')(P))$  [Préstamos]

Ahora para obtener a los que tienen depósitos pero no préstamos

$V4 \leftarrow V1 - V2$

## Ejemplo:

### Diferencia $R_1 - R_2$

Ejemplo: Los clientes de S5 que tienen depósito pero no préstamos

En SQL:

```
SELECT Idcliente  
FROM Deposito  
WHERE Idsucursal ='S5'
```

**MINUS**

```
SELECT Idcliente  
FROM Prestamo  
WHERE Idsucursal ='S5'
```

## Ejemplo:

### Intersección $R_1 \cap R_2$

Ejemplo: Los clientes que tienen depósito y préstamos en S5

$V1 \leftarrow \pi (\text{Idcliente}) (\sigma(\text{Idsucursal}='S5')(D))$  [Depósitos]

$V2 \leftarrow \pi (\text{Idcliente}) (\sigma(\text{Idsucursal}='S5')(P))$  [Préstamos]

Ahora para obtener a los clientes que tienen depósitos y préstamos en S5

$V5 \leftarrow V1 \cap V2$

## Ejemplo:

### Intersección $R_1 \cap R_2$

Ejemplo: Los clientes que tienen depósito y préstamos en S5

En SQL:

```
SELECT Idcliente  
FROM Deposito  
WHERE Idsucursal ='S5'
```

**INTERSECT**

```
SELECT Idcliente  
FROM Prestamo  
WHERE Idsucursal ='S5'
```

# El Modelo Relacional

## Restricciones de Integridad

Aseguran la consistencia de los datos. Se tienen las siguientes:

1. Integridad de dominio
2. Integridad de columnas
3. Integridad de entidad
4. Integridad referencial
5. Integridad definida por el usuario

# Restricciones de Integridad

## Integridad de dominio

El valor que se puede asignar a un dato debe estar en el dominio.

La excepción de la “información faltante” :

El modelo relacional permite a un dato estar marcado para no contener ningún valor del dominio. Algunos dicen que contiene el “valor **NULO**”.

Un dato puede estar marcado como NULO si :

- a) Se desconoce la información (NULO-APLICABLE). Permite NULO y valores de dominio.
- b) No tiene sentido asignar un valor del dominio (NULO-INAPLICABLE).

# Restricciones de Integridad

## Integridad de dominio

Una definición adecuada de integridad de dominio no sólo permitirá controlar valores insertados/actualizados sino controlar la coherencia de consultas (si las comparaciones en los predicados tienen sentido).

Ejemplo :

Entidad alumnos, columna número de libros a prestarse, debe tomar valores de los números naturales.

Atributo	Dominio	Descripción Dominio
nombre	NOMBRES	cadena de hasta 30 caracteres alfabéticos
telefono	TELEFONOS	cadena de hasta 9 caracteres numéricos
altura	MEDIDAS	números reales entre 0 y 2'5 (metros)
...	...	...

# Restricciones de Integridad

## Integridad de columnas

Dentro de la integridad de dominio puede adicionarse restricciones a nivel columna.

Ejemplo:  
entidad **Alumno**

columna cantidad de libros a prestarse menor a 5,  
columna número de cursos a matricularse menor a 18  
créditos.

En los DBMS:

- Los dominios se implementan con TIPOS DE DATOS
- Los subconjuntos de dominios se implementan con la integridad de columnas



# Restricciones de Integridad

## Integridad de entidad

Ningún componente de una llave primaria puede tener valores nulos.

Ningún componente de una llave foránea debe permitir un valor nulo por inaplicable.

En una base de datos relacional, nunca registraremos información acerca de algo que no podamos identificar.

# Restricciones de Integridad

## Integridad referencial

Para cada llave foránea debe existir un valor equivalente de una llave primaria y en el mismo dominio.

Ejemplo:

Si tenemos una entidad **direccion** con una llave foránea el código postal debe existir una entidad de **distrito** que tenga como llave primaria el código postal.

# Restricciones de Integridad

## Integridad definida por el usuario

Se permiten restricciones definidas por el usuario que pueden tener :

- aspectos de tiempo
- acciones de usuarios y de programas de aplicación
- condiciones de verificación
- acciones a tomarse en caso de violaciones.

### Ejemplo

- Una “regla de negocio” : llegar tarde al trabajo actualiza ...

# Agenda

- IDEF1X: modelamiento y convenciones
- Componentes
  - Entidades
  - Relaciones
  - Atributos

# IDEF1X: Técnica de modelamiento y convenciones

## ¿Qué es IDEF1X?

IDEF1X, es usado para realizar modelos de información gráfico que representan la estructura y semántica de la información dentro de un entorno o sistema. El uso de este estándar permite la construcción de un modelo de datos semántico, que podía servir para soportar la administración de datos como un recurso, la integración de sistemas de información y la construcción de base de datos computarizados.

**I**ntegration  
**DEF**inition  
**F**or  
**I**nformation  
**M**odeling

IDEF1X (IDEF1 Extended)

# IDEF1X: Técnica de modelamiento y convenciones

## El propósito

Esta técnica es usada para modelar los datos en un estándar, consistente, para administrarla de una manera predecible como un recurso. Los objetivos de este estándar son:

- Proveer un significado para entender y analizar completamente los recursos de datos de una organización.
- Proveer un significado común para representar y comunicar la complejidad de los datos.
- Proveer un método para representar una vista general de los datos requeridos en una organización.
- Proveer un significado para definir una vista de datos independiente de la aplicación el cual pueda ser validado por usuarios y transformado en un modelo de datos físico.
- Proveer un método para obtener una definición de datos integrados desde los recurso de datos existentes.

# IDEF1X: Técnica de modelamiento y convenciones

## El producto

Un modelo de datos:

- Varias proyecciones: “vistas funcionales”
- Entidades de datos: todas con llaves
- Relaciones entre entidades: sin datos, binarias y especificadas (no relaciones de muchos a muchos).
- Integridad existencial: llaves primarias
- Integridad referencial: llaves foráneas
- Estructuras de clasificación: exclusivas
- Las entidades se implementarán como tablas relacionales

# IDEF1X: Técnica de modelamiento y convenciones

## Sintaxis y Semántica

Los componentes de una vista IDEF1X son:

### 1) Entidades

Independientes

Dependientes

### 2) Relaciones

Identificadora

No-Identificadora

De Categoría

No-especifica

### 3) Atributos

Atributo/Clave

Atributo

Clave Primaria

Clave Alterna

Clave Foránea



# IDEF1X: Técnica de modelamiento y convenciones

## 1. Entidades

### Entidad Independiente

Cada instancia de la entidad puede ser únicamente identificada sin determinar su relación con otra entidad



Entidad Padre

### Entidad Dependiente

La única identificación de una instancia de la entidad depende de su relación con otra entidad



Entidad Hijo

# IDEF1X: Técnica de modelamiento y convenciones

## 1. Entidades

### Entidad Independiente

Solo una parte de una llave foránea o ninguna llave foránea es usada como parte de la llave primaria.

Entidad Padre AtribPrimo1 AtribPrimo2
etc..

### Entidad Dependiente

La llave foránea (completa) es usada en toda o parte de la llave.

Entidad Hijo AtribPrimo Llave Foránea
etc.

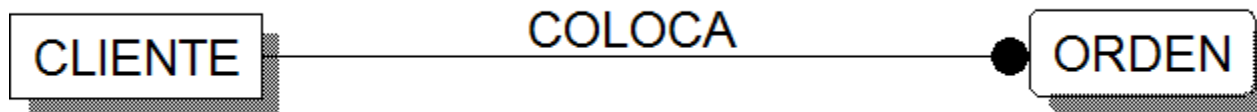
# IDEF1X: Técnica de modelamiento y convenciones

## 2. Relaciones

Son usadas para representar las asociaciones entres entidades.

Para cuantificar cualquier tipo de asociación, se utiliza la cardinalidad.

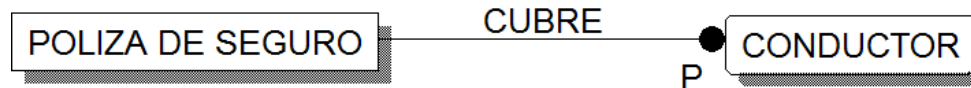
Desde la perspectiva de la entidad padre, se pueden ver las siguientes cardinalidades:



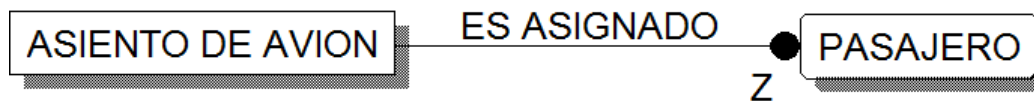
Cada instancia del padre esta relacionada con cero, uno o más instancias del hijo.

# IDEF1X: Técnica de modelamiento y convenciones

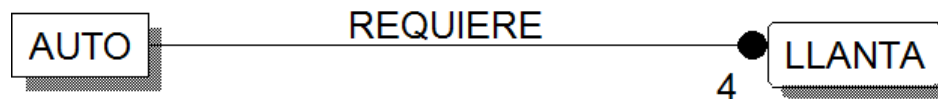
## Cardinalidad de las Relaciones



Cada instancia del padre esta relacionada con una o más instancias del hijo.



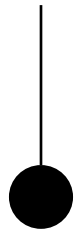
Cada instancia del padre esta relacionada con cero o una instancia del hijo.



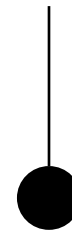
Cada instancia del padre esta relacionada exactamente con “N” instancias del hijo.

# IDEF1X: Técnica de modelamiento y convenciones

## Cardinalidad de las Relaciones



Cero, uno o más

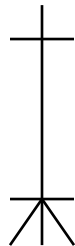


**n**

Exactamente n



**P**

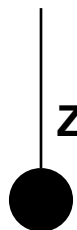


Uno o más

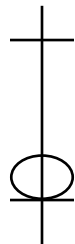


**m-M**

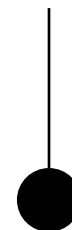
mínimo m y  
máximo M



**Z**



Cero o uno



**(n)**

Ver nota (n)

# IDEF1X: Técnica de modelamiento y convenciones

## 2. Relaciones

### Relación Identificadora

Una instancia de la entidad hijo está identificada por la asociación con la entidad padre.

“la entidad hijo solo existe si existe el padre”

### Relación No-identificadora

Cada instancia de la entidad hijo puede ser unívocamente identificada sin conocer la instancia asociada de la entidad padre.

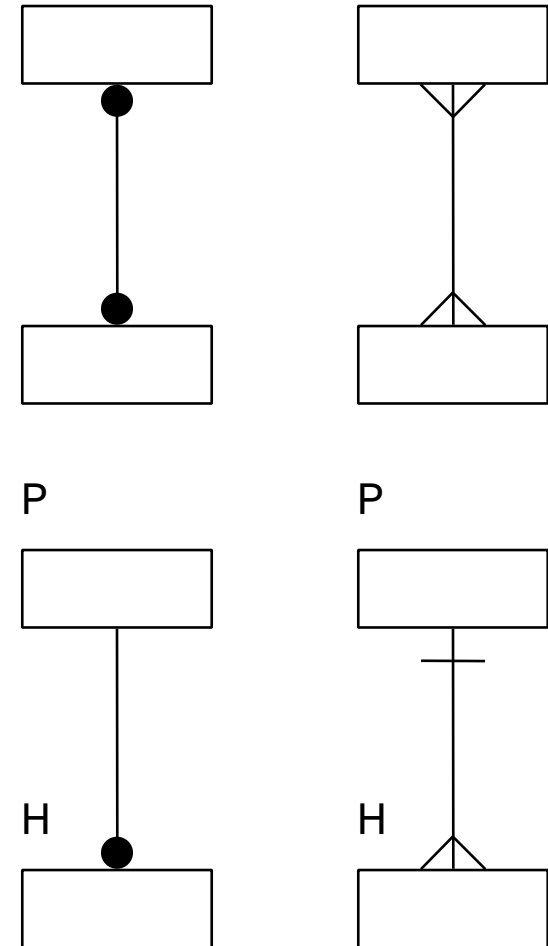
a) Mandatoria. Cada instancia de la entidad hijo está relacionada exactamente con una instancia de la entidad padre.

b) No mandatoria u opcional. Cada instancia de la entidad hijo está relacionada con cero o una instancia de la entidad padre.

# IDEF1X: Técnica de modelamiento y convenciones

## Tipos de Relaciones (cardinalidades máximas)

- **Relaciones No Especificadas**  
M-M o “Muchos a Muchos”  
Las cardinalidades máximas (de ambos sentidos en la relación binaria) son mayores que 1.
- **Relaciones Especificadas**  
Las cardinalidad máxima en un sentido es 1.  
Este sentido es de “Hijo” a “Padre”



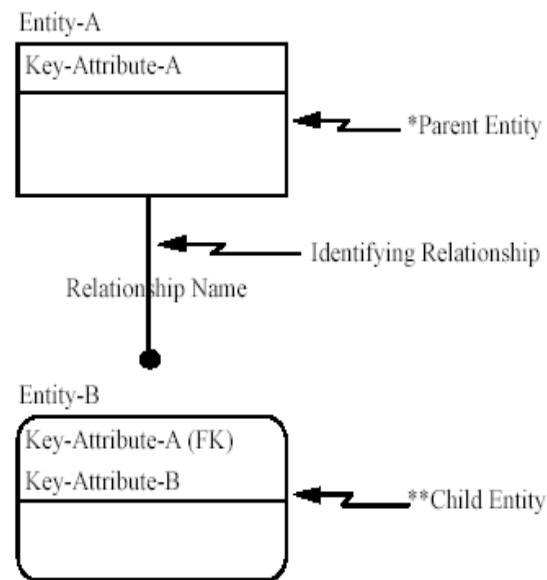
# IDEF1X: Técnica de modelamiento y convenciones

## 2. Relaciones

### Relación Identificadora

Una instancia de la entidad hijo está identificada por la asociación con la entidad padre.

“la entidad hijo solo existe si existe el padre”



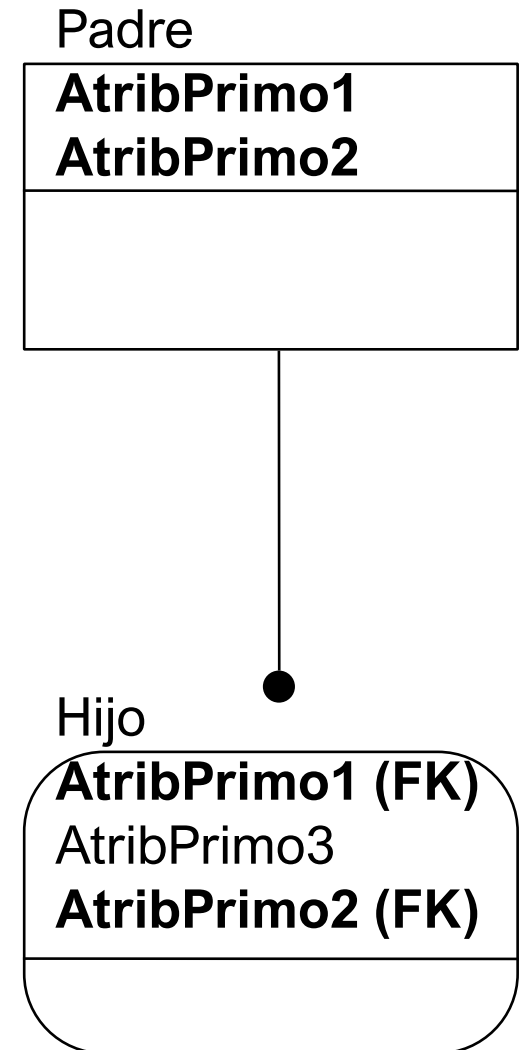


# IDEF1X: Técnica de modelamiento y convenciones

## 2. Relaciones

### Relación Identificadora

- Una llave foránea (llave primaria completa de la entidad “Padre”) es usada en toda o parte de la llave primaria.
- La entidad “Hijo” siempre será dependiente (dependencia existencial)
- La entidad “Padre” puede ser independiente o dependiente de otra entidad.



## 2. Relaciones

### Relación No-identificadora

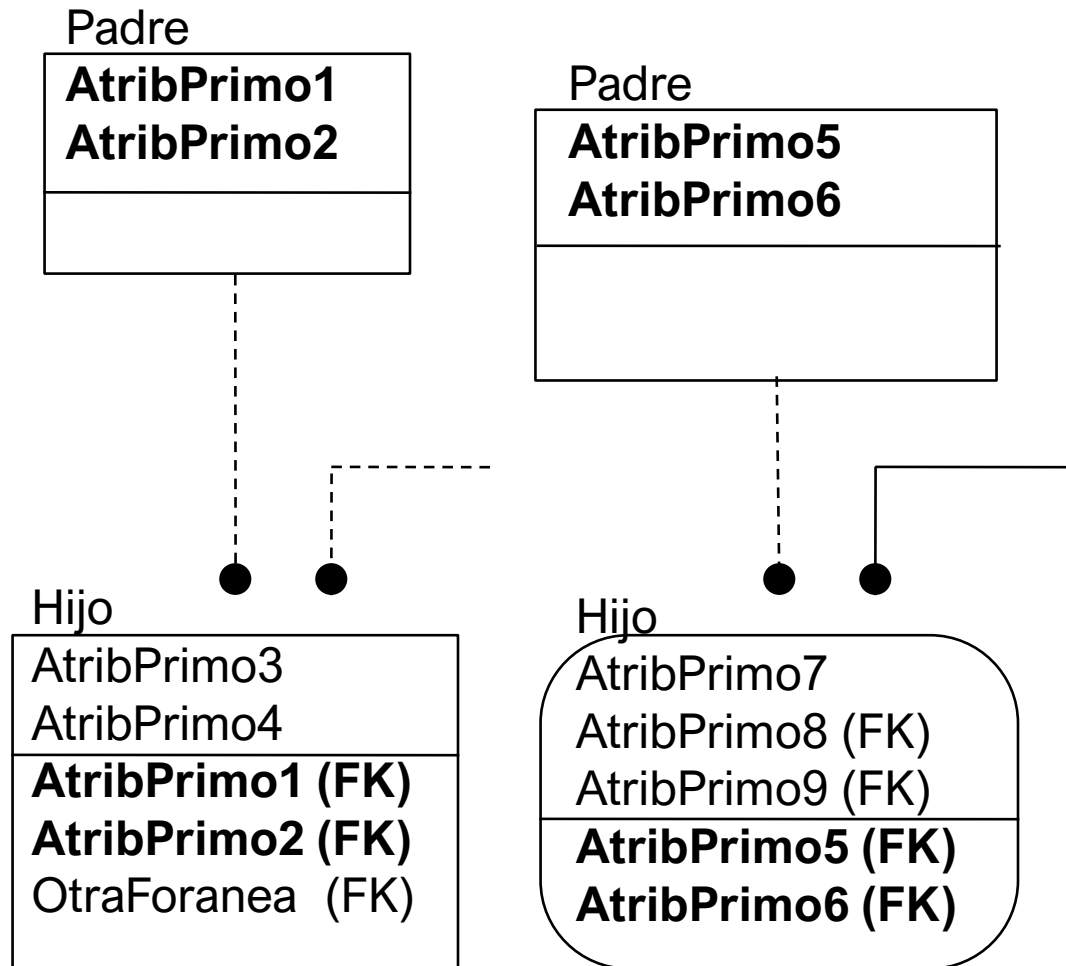
Cada instancia de la entidad hijo puede ser unívocamente identificada sin conocer la instancia asociada de la entidad padre.

- a) **Mandatoria.** Cada instancia de la entidad hijo está relacionada exactamente con una instancia de la entidad padre.
- b) **No mandatoria u opcional.** Cada instancia de la entidad hijo está relacionada con cero o una instancia de la entidad padre.

## Relación No Identificada (Referencia)

Si cada instancia de la entidad hijo puede ser identificado de manera única, independientemente de la asociación con la entidad padre.

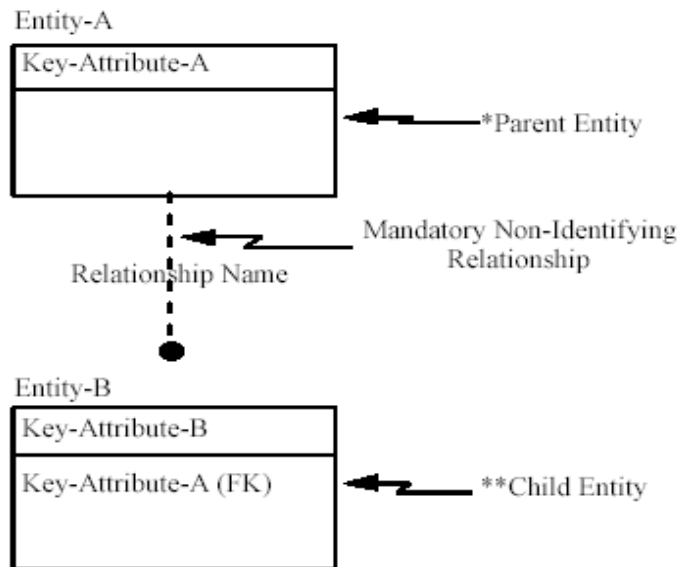
- Una llave foránea (llave primaria completa de la entidad “Padre”) es usada como atributo(s) en la entidad “Hijo”.
- Tanto el “Padre” como el “Hijo” podrán ser Independientes o Dependientes, en función a sus relaciones con otras entidades



# Relación No Identificada (Referencia)

## Relación Mandatoria

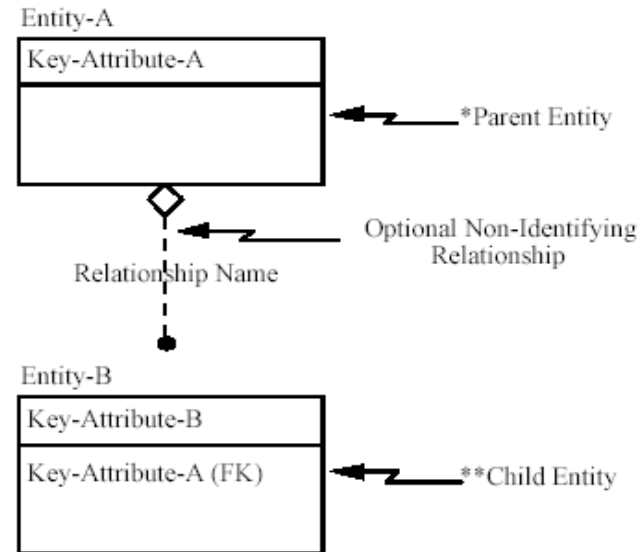
La cardinalidad mínima de relación Hijo-Padre es 1.



Cada instancia de la entidad hijo corresponde a exactamente una instancia del padre. Llave foránea (FK) NO NULA

## Relación Opcional

La cardinalidad mínima de relación Hijo-Padre es 0.



Cada instancia de la entidad hijo esta relacionada a cero o una instancia de la entidad padre. Llave foránea (FK) admite NULOS

## 2. Relaciones

### Relación de Categoría

Es usada para representar estructuras en las cuales una entidad es un tipo de categoría de otra entidad.

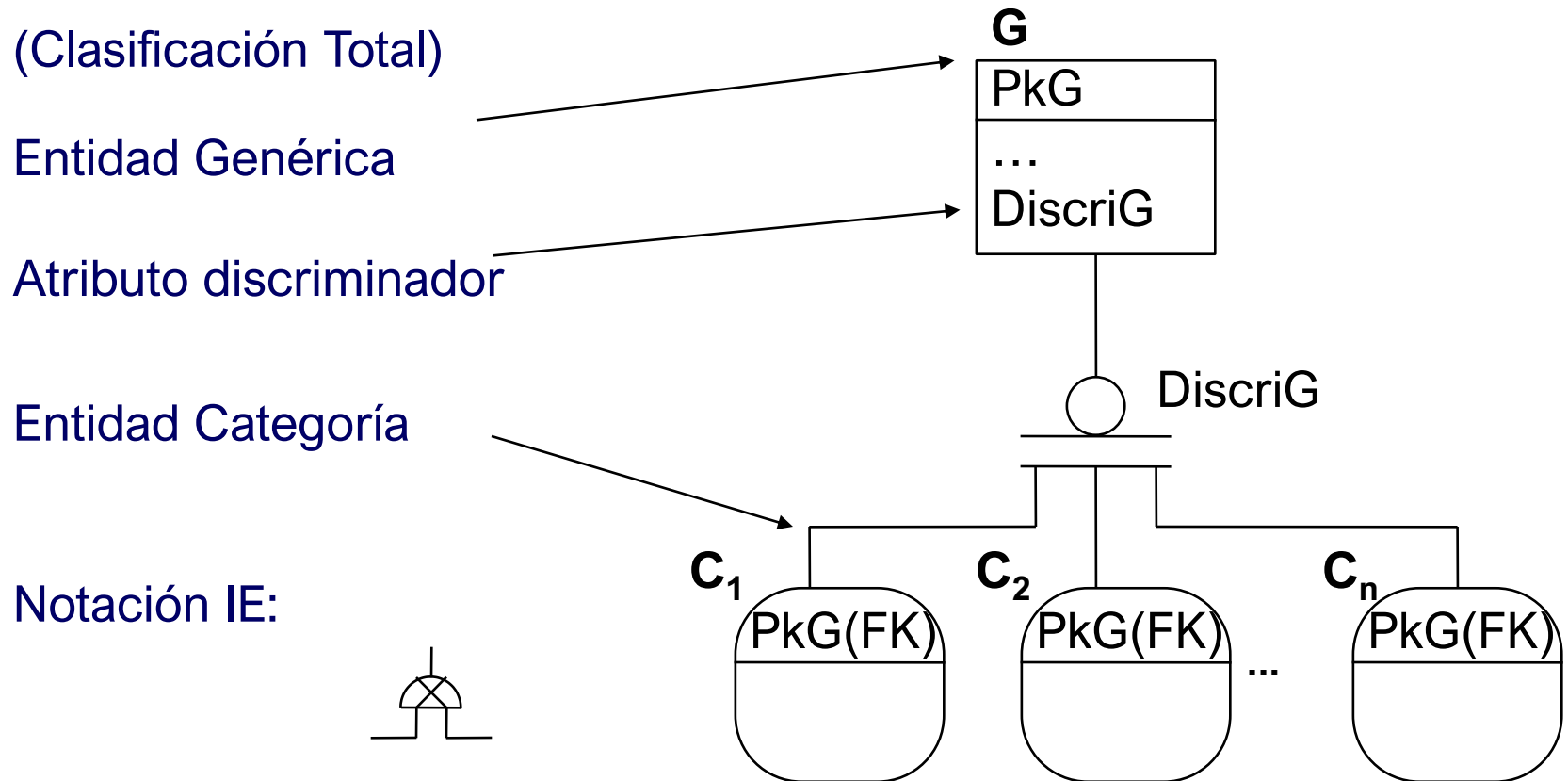
La cardinalidad no se especifica, ya que siempre es cero o uno. Las entidades categorías son siempre dependientes-identificadas.

Tipos de relaciones de categorización

- Categorización Completa
- Categorización Incompleta

# IDEF1X: Técnica de modelamiento y convenciones

## Relación de Categorización Completa



# IDEF1X: Técnica de modelamiento y convenciones

## Relación de Categorización Incompleta

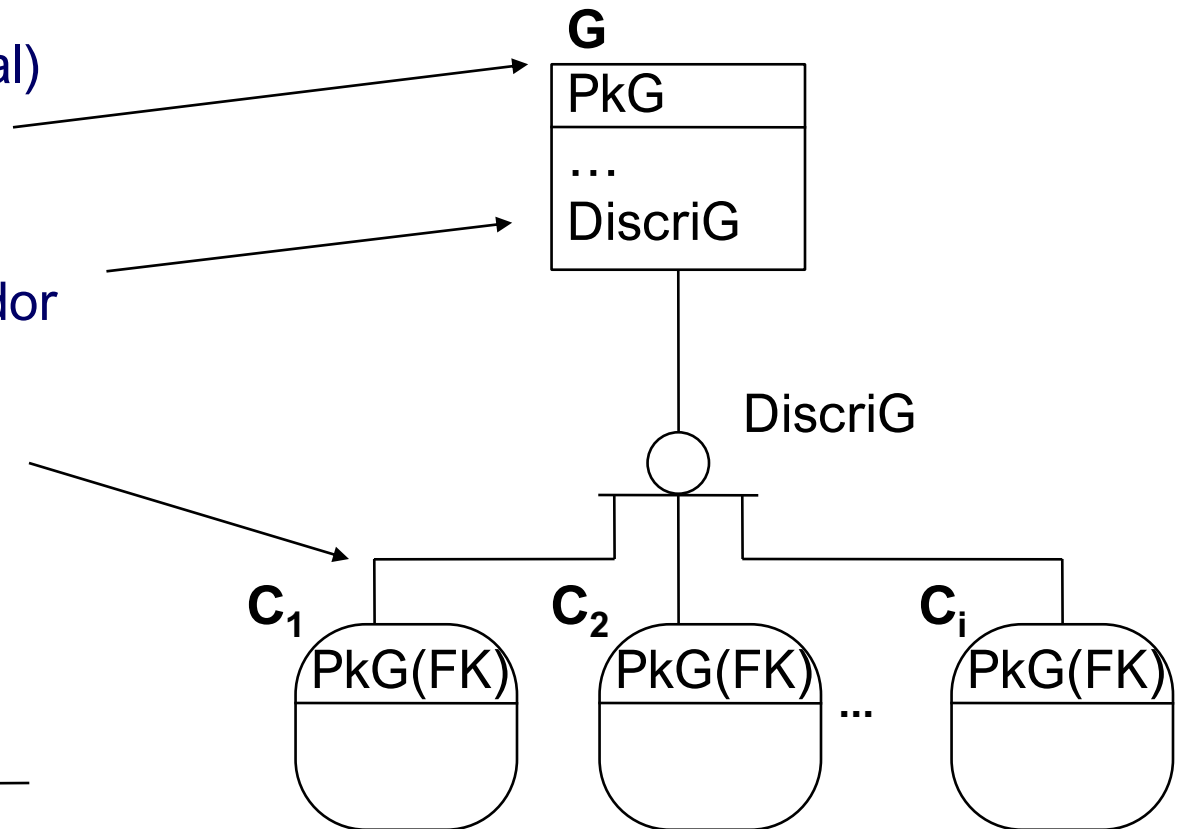
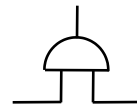
(Clasificación Parcial)

Entidad Genérica

Atributo discriminador

Entidad Categoría

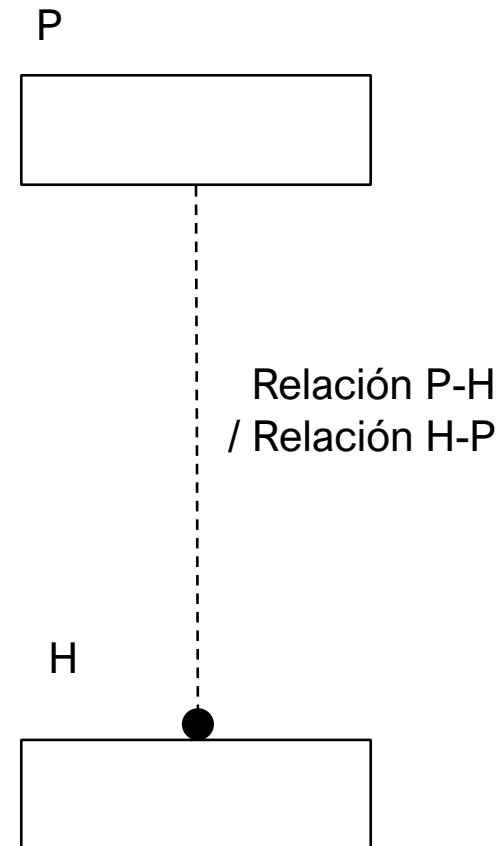
Notación IE:



# IDEF1X: Técnica de modelamiento y convenciones

## Nombre(s) de la Relación

- El nombre hará referencia al sentido “Padre” a “Hijo”.
- El nombre de la relación inversa (sentido “Hijo” a “Padre”) va precedida por un slash ( / ).
- Puede indicarse el nombre de la relación, el de la inversa o ambas.





# IDEF1X: Técnica de modelamiento y convenciones

## 2. Relaciones

### Relación No-específica

Es usada en un alto nivel del Diagrama Entidad-Relación para representar relaciones **m:n** entre entidades.

Las relaciones de categoría son consideradas relaciones específicas, ya que ellas definen precisamente cuantas instancias de una entidad se relacionan con las instancias de otra entidad.

# IDEF1X: Técnica de modelamiento y convenciones

## 3. Atributos

Representan las características distintivas de una entidad y de las cuales se almacena información.

### Clave Primaria

Un atributo o grupo de atributos que identifican unívocamente una instancia de una entidad.

### Clave Alterna

Un atributo o conjunto de atributos que identifican unívocamente cada instancia, pero no forman parte de la clave primaria.

### Clave Foránea

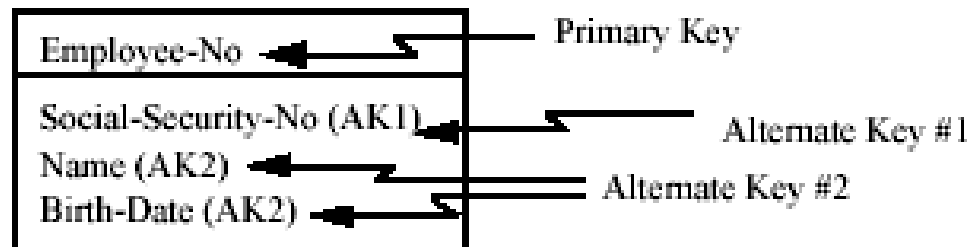
Una clave primaria de una entidad padre que es aportada a una entidad hijo, a través de una relación.

# IDEF1X: Técnica de modelamiento y convenciones

## Llaves Primarias (PK) y Alternas (AK)

- Representan restricciones de unicidad sobre los valores de los atributos de la entidad
- Llave candidata es uno o más atributos cuyos valores identifican de manera única a la instancia de una entidad
- En algunos casos, una entidad podría tener más de un atributo o grupo de atributos que lo identifican de manera única. Si existe más de una llave candidata, una de ellas es designada como llave primaria y las otras como llaves alternas

Example



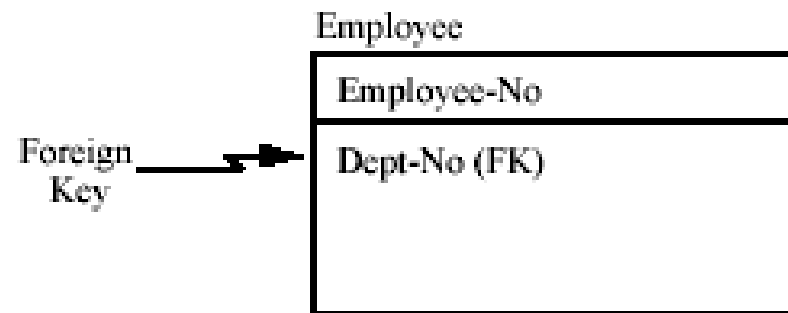
## Llaves Foráneas (FK)

- Si existe una “relación” entre dos entidades, los atributos que conforman la llave primaria del padre o entidad genérica son migrados como atributos del hijo o categoría de la entidad (en caso de una categorización). Estos atributos son llamados llaves foráneas (FK)
- Los atributos migrados podrían ser usados como:
  - Una parte o toda la llave primaria (entidad dependiente, categoría)
  - Llave alterna
  - Atributo no llave (entidad independiente)

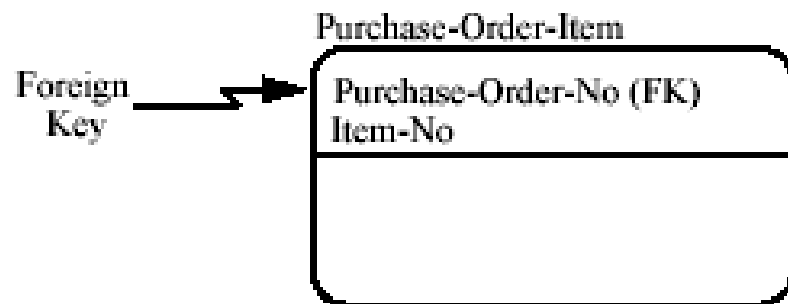
# IDEF1X: Técnica de modelamiento y convenciones

## Llaves Foráneas (FK)

Migrated Non-key Attribute Example



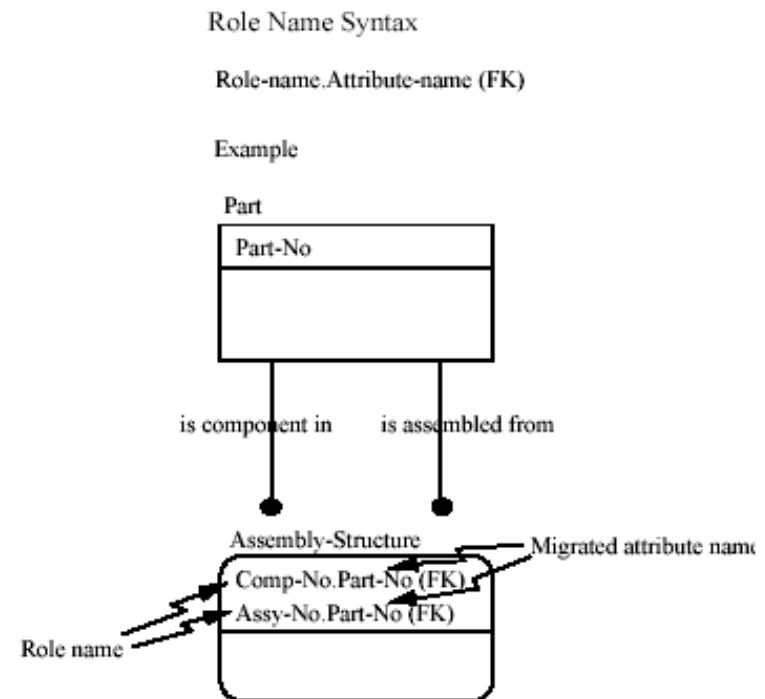
Migrated Primary Key Attribute Example



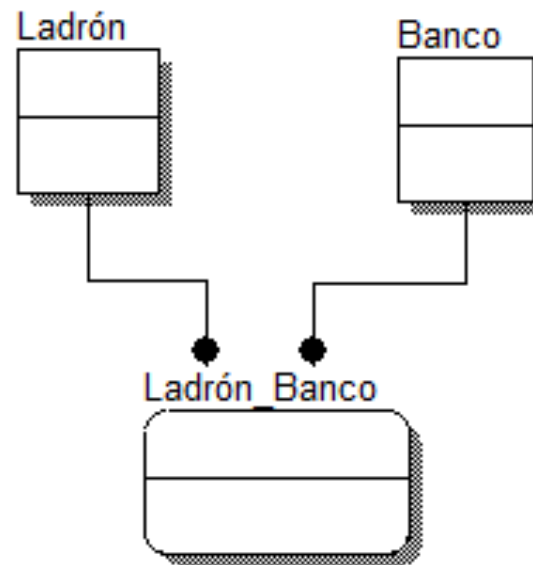
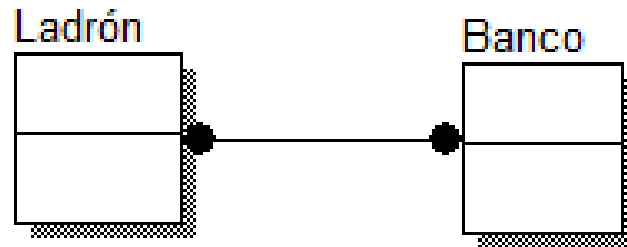
# IDEF1X: Técnica de modelamiento y convenciones

## Rolename

- Cuando un atributo es migrado a una entidad a través de más de una relación, se hace necesario un *rolename* para asignarlo a cada ocurrencia y diferenciarlo de los demás.
- Si para una instancia los atributos pueden tener diferentes valores, cada atributo migrado debe tener un *rolename* diferente, si por otro lado, deben tener el mismo valor, deben tener el mismo *rolename*

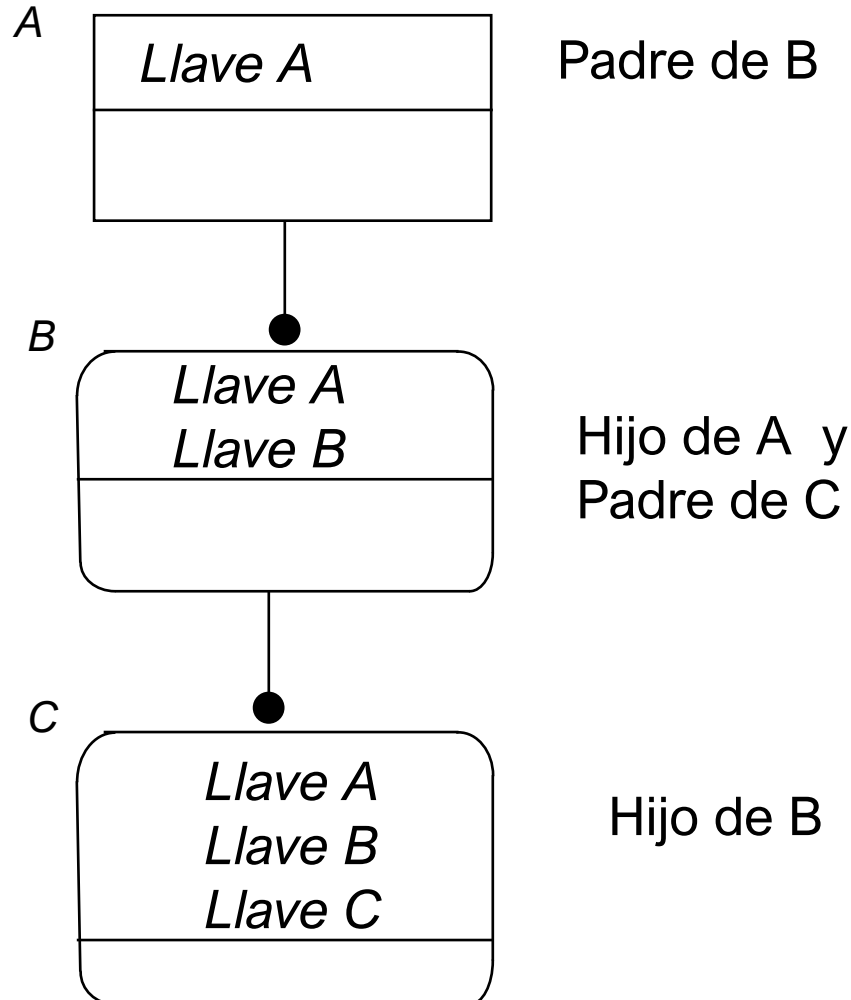


# Solución a Relaciones No Especificadas



# Llaves y Especificación de Relaciones

## Jerarquía de identificación





En esta sesión, debe haber aprendido lo siguiente:

- Comprender el Modelo Relacional
- Comprender Algebra relacional

Adicional

## Notación Richard Barker

# Notación Barker

La siguiente notación fue desarrollada originalmente por la consultora británica CACI. Posteriormente, fue promovido por Richard Barker y adoptado por *Oracle Corporation* para su "método CASE\*" (posteriormente rebautizado como "método de desarrollo personalizado").

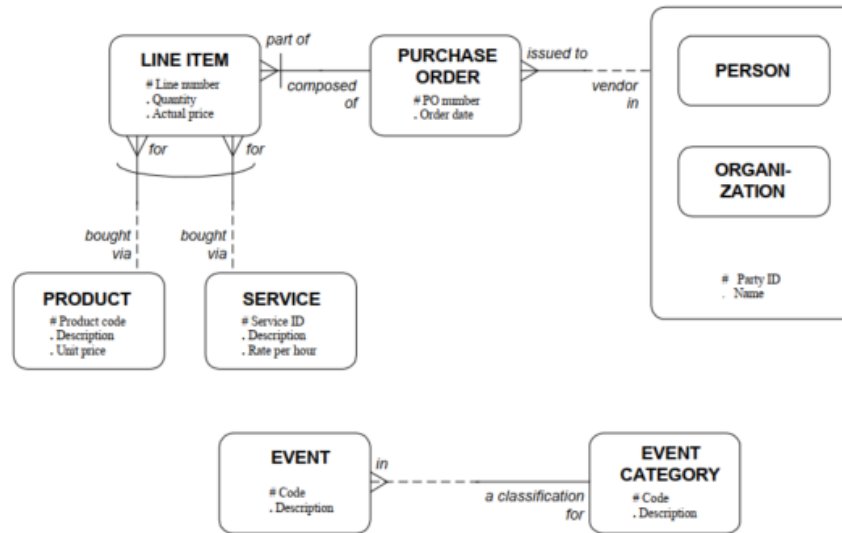


Figura. Un modelo de datos del método CASE\*

# Notación Barker

## Convenciones de Diagramas de ER

- Los nombres de entidades van en los cuadros editables.
- Los nombres de entidades están siempre en singular y escritos con todas las letras mayúsculas.

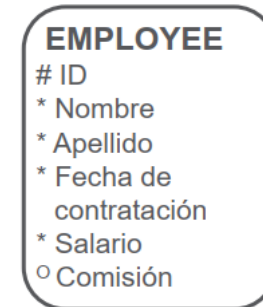
**EMPLEADO**

**JOB**

# Notación Barker

## Convenciones de Diagramas

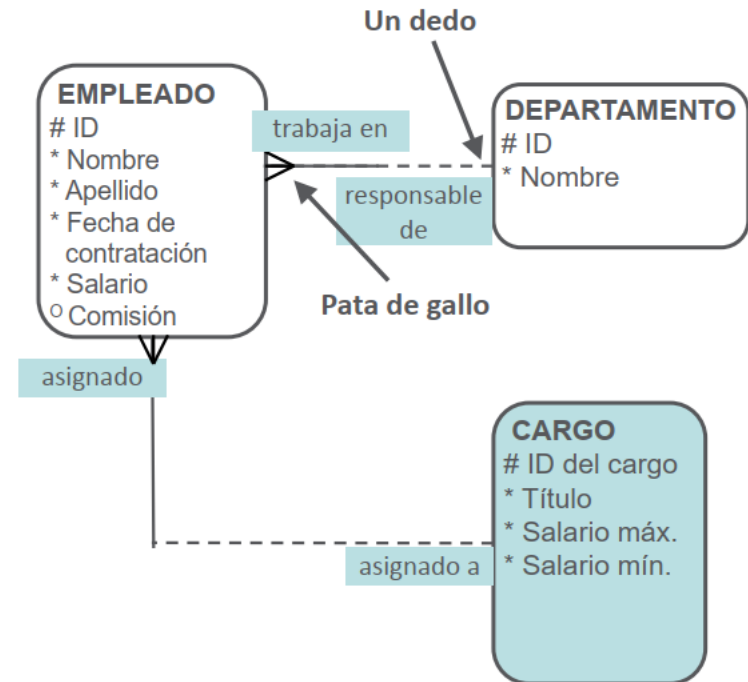
- Los atributos se muestran bajo los nombres de entidades.
- Los atributos obligatorios están marcados con un asterisco: "\*"
- Los atributos opcionales están marcados con un círculo: "o"
- Los identificadores únicos están marcados con un símbolo de almohadilla: "#"



# Notación Barker

## Convenciones de Diagramas

- Las relaciones son líneas que conectan entidades.
- Estas líneas son continuas o discontinuas.
- Estas líneas terminan en un "dedo" o "pata de gallo" al final de cada entidad



# Notación Barker

## Cardinalidad de las Relaciones

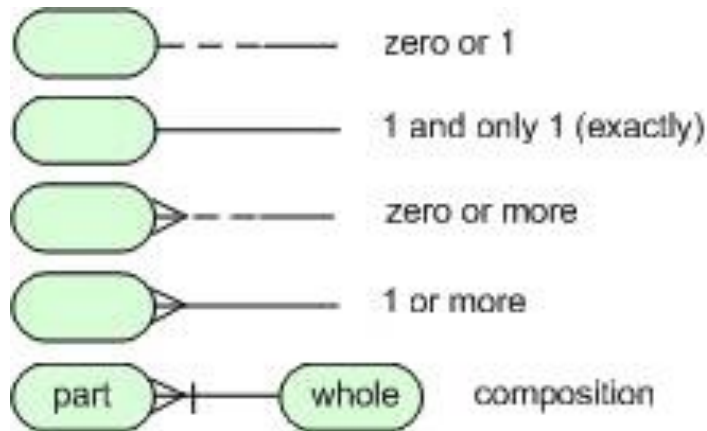


Figura. Notación Richard Barker

# Notación Barker

## Componentes de ERD

- CADA
- Entidad A
- OPCIONALIDAD (debe ser/puede ser)
- NOMBRE DE RELACIÓN
- CARDINALIDAD (solo uno/uno o más)
- Entidad B



# Notación Barker

## Componentes de ERD

- Puesto que cada relación tiene dos partes, leemos la primera relación de izquierda a derecha (o de arriba a abajo, según el diseño del ERD)



Figura. ERD. Fuente: ORACLE Academy

# Notación Barker

## Componentes de ERD

- Ahora leeremos la relación de derecha a izquierda.

1. CADA
2. Entidad B
3. OPCIONALIDAD (debe ser/puede ser)
4. NOMBRE DE RELACIÓN
5. CARDINALIDAD (solo uno/uno o más)
6. Entidad A

### Desglose de ERD



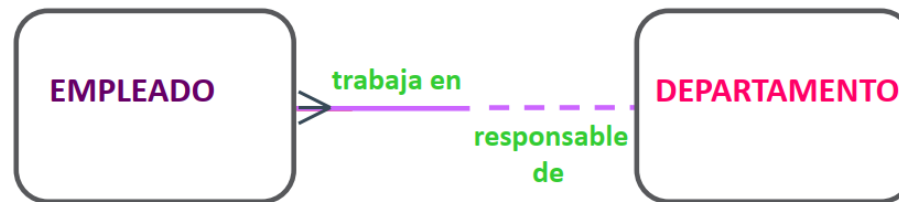
1. CADA
2. **DEPARTAMENTO** (entidad B)
3. **PUEDE SER** (opcionalidad, línea discontinua)
4. **RESPONSABLE DE** (nombre de relación)
5. **UNO O MÁS** (cardinalidad, pata de gallo)
6. **EMPLEADO** (entidad A)

Figura. ERD. Fuente: ORACLE Academy

# Notación Barker

## Componentes de ERD

- Ahora reúnalo todo.



1. CADA
2. **EMPLEADO** (entidad A)
3. **DEBE** (opcionalidad, línea continua)
4. **TRABAJA EN** (nombre de relación)
5. **SOLO UNO** (cardinalidad, un dedo)
6. **DEPARTAMENTO** (entidad B)

1. CADA
2. **DEPARTAMENTO** (entidad B)
3. **PUEDE SER** (opcionalidad, línea discontinua)
4. **RESPONSABLE DE** (nombre de relación)
5. **UNO O MÁS** (cardinalidad, pata de gallo)
6. **EMPLEADO** (entidad B)

Figura. ERD. Fuente: ORACLE Academy

# Conclusiones

Se han revisado los principales conceptos del modelo relacional como son:

- Estructura
- Operaciones relacionales
- Restricciones de integridad

Además técnicas de modelamiento de datos como:

- IDEF1X
- Barker