

# Cálculo y Álgebra Relacional

(Cap 6 - Elmasri 5ª edición)

# Introducción

- ♦ Vamos a tratar los dos lenguajes formales del modelo relacional.
  - Cálculo relacional
  - Álgebra relacional
- ♦ El **modelo de datos** debe incluir un conjunto de **operaciones** para **manipular** la BD.
- ♦ Estas operaciones permiten al usuario especificar las peticiones fundamentales de **recuperación de información en la BD**.
- ♦ El resultado de una recuperación es una nueva relación, y estas relaciones se pueden manipular posteriormente.
- ♦ El conjunto de **operaciones básicas del modelo relacional** es lo que se denomina **álgebra relacional** (AR).
- ♦ El AR proporciona el **fundamento formal para las operaciones del modelo relacional**, y además es la **base para la implementación y optimización** de las consultas en un sistema de gestión de BD.

# Introducción

- ♦ El **cálculo relacional** (CR) ofrece una **notación declarativa** de alto nivel para especificar las consultas relacionales.
- ♦ En una expresión de cálculo no existe un orden de las operaciones para recuperar los resultados de la consulta: la expresión solo especifica la información que el resultado debería contener.
- ♦ Por tanto el **CR es un formalismo declarativo, al contrario del AR que es procedural** (especifica el conjunto de operaciones en el orden adecuado para recuperar los resultados de la consulta).
- ♦ Así tenemos dos tipos de formalismos para expresar operaciones de consulta sobre una BD en el modelo relacional, **CR y AR**.
- ♦ Estos dos formalismos **son diferentes pero lógicamente equivalentes**:
  - Toda expresión de cálculo se puede expresar en álgebra y viceversa.
  - Es decir, permiten expresar las mismas consultas.

# Introducción

- ♦ Un lenguaje de consulta es **relacionalmente completo** si permite expresar cualquier consulta del cálculo relacional.
- ♦ La traducción de AR a SQL es relativamente inmediata.
- ♦ Los **motores de SQL** basan su representación interna en consultas a través de **AR**.
- ♦ Así el **AR se utiliza con fines más prácticos**, en principio es más manejable que el SQL para diseñar consultas complejas.
- ♦ La traducción de CR a SQL no es trivial.
- ♦ **CR es más adecuado para establecer y verificar propiedades formales, la consistencia de los modelos relacionales y sus formalismo.**
- ♦ Introducido por Edgar Frank **Codd**, para operar en el contexto del modelo relacional.
- ♦ La **creación original del modelo relacional se fundamentó en el cálculo** —interesa entenderlo para una comprensión más profunda del modelo relacional y el fundamento de la tecnología de bases de datos y otros lenguajes de consulta de BDs como **QBE** (desarrollado por IBM).

# Introducción

- ♦ QBE: (**Query By Example** - Consulta por ejemplo).
- ♦ Es un método de consulta en base de datos relacionales. Fue ideado por Moshé M. Zloof en el IBM Research a mediados de los 70, en paralelo al desarrollo de SQL.
- ♦ Este sistema nos permite que la persona que genera la búsqueda o la aplicación que la realice pueda **proporcionar información sobre aquello que está buscando en la base de datos**.
- ♦ Fue el **primer lenguaje de consulta gráfico**, que utiliza tablas visuales donde el usuario puede insertar comandos, elementos de ejemplos y condiciones.
- ♦ QBE está basado en la idea de DRC (Domain relational calculus), que es un cálculo que fue introducido por Michel Lacroix y Alain Pirotte como un lenguaje de consultas declarativo de base de datos para el modelo relacional.
- ♦ Más información en: [http://es.wikipedia.org/wiki/Búsqueda\\_mediante\\_ejemplo](http://es.wikipedia.org/wiki/Búsqueda_mediante_ejemplo)
- ♦ Esto es interesante por **Page Rank**, es QBE con texto: consiste en dar al sistema cualquier información y que éste devuelva entradas relacionadas con el tema

# Cálculo Relacional

- ♦ El CR es un lenguaje formal basado en una rama de la lógica matemática llamada cálculo de predicado de primer orden.
- ♦ El CR puede actuar sobre tuplas o dominios (diferentes tipos de variables):
  - CRT
  - CRD
- ♦ El CR de tupla opera directamente sobre las tuplas de una relación, las variables utilizadas en la fórmulas operan sobre tuplas.
- ♦ En el CR de dominio las variables usadas en las fórmulas operan sobre valores individuales de los dominios de los atributos.
- ♦ La diferencia es esencialmente de notación, pero son prácticamente equivalentes.

# Cálculo Relacional de Tupla

- ♦ El CR de tupla está basado en la especificación de un número de variables de tuplas.
- ♦ Cada especificación suele aplicarse sobre una relación base de datos en particular:
  - Es decir la variable podría tomar su valor de cualquier tupla individual de esa relación base
- ♦ Un consulta de CR sencilla tiene la siguiente forma  $\{ t \mid \textit{cond}(t) \}$ :
  - $t$  representa una variable de tupla
  - $\textit{cond}(t)$  es una expresión condicional
  - La expresión representa (literalmente) un conjunto de tuplas que cumplen la condición
- ♦ El resultado es el conjunto de todas las tuplas  $t$  que satisfacen la condición  $\textit{cond}(t)$ .

# BD Empresa

## EMPLEADO

▲	Nombre text	Apellido1 text	Apellido2 text	Dni integer	FechaNac date	Direccion text	Sexo "char"	Sueldo numeric	SuperDni integer	Dno integer
1	Jose	Perez	Perez	123456789	1965-09-01	Eloy I, 98	H	30000	333445555	5
2	Alberto	Campos	Sastre	333445555	1955-12-08	Avda Rios, 9	H	40000	888665555	5
3	Alicia	Jimenez	Celaya	999887777	1968-05-12	Gran Via, 38	M	25000	987654321	4
4	Juana	Sainz	Oreja	987654321	1941-06-20	Cerquillas, 67	M	43000	888665555	4
5	Eduardo	Ochoa	Paredes	888665555	1937-11-10	Las Peñas, 1	H	55000	[null]	1
6	Fernan...	Ojeda	Ordoñez	666884444	1962-09-15	Portillo, S/N	M	38000	333445555	5
7	Luis	Pajares	Morera	987987987	1969-03-29	Enebros, 90	H	25000	987654321	4
8	Aurora	Oliva	Avezuela	453453453	1972-07-31	Anton, 6	M	25000	333445555	5

## TRABAJA\_EN

▲	DniEmpleado integer	NumProy integer	Horas numeric
1	123456789	1	32.5
2	123456789	2	7.5
3	666884444	3	40.0
4	453453453	1	20.0
5	453453453	2	20.0
6	333445555	2	10.0
7	333445555	10	10.0
8	333445555	3	10.0
9	333445555	20	10.0
10	999887777	30	30.0
11	999887777	10	10.0
12	987987987	10	35.0
13	987987987	30	5.0
14	987654321	30	20.0
15	987654321	20	15.0
16	888665555	20	[null]

## PROYECTO

▲	NombreProyecto text	NumProyecto integer	UbicacionProyecto text	NumDptoProyecto integer
1	PruductoX	1	Valencia	5
2	ProductoY	2	Sevilla	5
3	ProductoZ	3	Madrid	5
4	Computacion	10	Gijon	4
5	Reorganizacion	20	Madrid	1
6	Comunicaciones	30	Gijon	4

## LOCALIZACIONES\_DPTO

▲	NumeroDpto integer	UbicacionDpto text
1	1	Madrid
2	4	Gijon
3	5	Valencia
4	5	Sevilla
5	5	Madrid

## SUBORDINADO

▲	DniEmpleado integer	NombSubordinado text	Sexo "char"	FechaNac date	Relacion text
1	333445555	Alicia	M	1986-04-05	Hija
2	333445555	Teodoro	H	1983-10-25	Hijo
3	333445555	Luisa	M	1958-05-03	Esposa
4	987654321	Alfonso	H	1942-01-28	Esposo
5	123456789	Miguel	H	1988-01-04	Hijo
6	123456789	Alicia	M	1988-12-30	Hija
7	123456789	Elisa	M	1967-05-05	Esposa

## DEPARTAMENTO

▲	NombreDpto text	NumeroDpto integer	DniDirector integer	FechaIngresoDirector date
1	Investigacion	5	333445555	1988-05-22
2	Administracion	4	987654321	1995-01-01
3	Sede Central	1	888665555	1981-06-19



# Campos, Tuplas y Tablas en una BD

## EMPLEADO

Nombre	Apellido1	Apellido2	<u>Dni</u>	FechaNac	Direccion	Sexo	Sueldo	SuperDni	Dno
text	text	text	integer	date	text	"char"	numeric	integer	integer

## DEPARTAMENTO

NombreDpto	<u>NumeroDpto</u>	<u>DniDirector</u>	FechaIngresoDirector
text	integer	integer	date

## LOCALIZACIONES\_DPTO

<u>NumeroDpto</u>	<u>UbicacionDpto</u>
integer	text

## PROYECTO

NombreProyecto	<u>NumProyecto</u>	UbicacionProyecto	NumDptoProyecto
text	integer	text	integer

## TRABAJA\_EN

<u>DniEmpleado</u>	<u>NumProy</u>	Horas
integer	integer	numeric

## SUBORDINADO

<u>DniEmpleado</u>	<u>NombSubordinado</u>	Sexo	FechaNac	Relacion
integer	text	"char"	date	text

# Cálculo Relacional de Tupla: Variables de Tupla y Relaciones de Rango

- ♦ Por ejemplo para localizar todos los empleados cuyo salario es superior a 50000€ podemos escribir la siguiente expresión:
  - $\{ t \mid \text{EMPLEADO}(t) \text{ AND } t.\text{sueldo} > 50000 \}$
  - La condición  $\text{EMPLEADO}(t)$  especifica que *la relación de rango* de la variable de tupla  $t$  es  $\text{EMPLEADO}$ .
  - Cada  $\text{EMPLEADO } t$  que satisface la condición  $t.\text{sueldo} > 50000$  será recuperada.
  - $t.\text{Sueldo}$  hace referencia al atributo  $\text{Sueldo}$  de la variable de tupla  $t$ .
- ♦ La consulta anterior recupera todos los valores del atributo de cada tupla. Para recuperar solo algunos atributos (Nombre y Apellido1):
  - $\{ t.\text{Nombre}, t.\text{Apellido1} \mid \text{EMPLEADO}(t) \text{ AND } t.\text{sueldo} > 50000 \}$
- ♦ Por ejemplo para localizar todos los vuelos con origen en Madrid
  - $\{ t \mid \text{Vuelo}(t) \text{ and } t.\text{origen} = \text{'MAD'} \}$
  - La condición  $\text{Vuelo}(t)$  especifica que *la relación de rango* de la variable de tupla  $t$  es  $\text{Vuelo}$ .
  - Cada  $\text{Vuelo } t$  que satisface la condición  $t.\text{origen} = \text{'MAD'}$  será recuperada.
  - $t.\text{origen}$  hace referencia al atributo  $\text{origen}$  de la variable de tupla  $t$ .

# Cálculo Relacional de Tupla: Consulta General

- ♦ Informalmente tenemos que suministrar la siguiente información en una expresión de cálculo de tupla:
  - Para cada variable de tupla  $t$ , la relación de rango  $R$  de  $t$ ,  $R(t)$ .
  - Una condición para seleccionar combinaciones de tuplas particulares.
  - El conjunto de atributos a recuperar (atributos solicitados).
- ♦ Por ejemplo recuperar la fecha de nacimiento y la dirección del empleado (o empleados) cuyo nombre sea José Pérez Pérez:
  - $\{ t.FechaNac, t.Dirección \mid EMPLEADO(t) \text{ AND } t.Nombre = 'José' \text{ AND } t.Apellido1 = 'Pérez' \text{ AND } t.Apellido2 = 'Pérez' \}$
- ♦ Pero se pueden formular consultas más elaboradas
- ♦ Vamos a ver la forma general de una consulta de CR:

$\{ \textit{variables} \mid \textit{condición} \}$

# Cálculo Relacional de Tupla: Expresiones y Fórmulas

- ♦  $\{ t_1 . A_j, t_2 . A_k, \dots, t_n . A_m \mid \text{COND}(t_1, t_2, \dots, t_n, t_{n+1}, t_{n+2}, \dots, t_{n+m}) \}$ 
  - Donde  $t_1, t_2, \dots, t_n, t_{n+1}, t_{n+2}, \dots, t_{n+m}$  son variables de tupla.
  - $A_j$  es un atributo de la relación a la que engloba  $t_i$
  - Y COND es una condición o fórmula.
- ♦ Una fórmula está compuesta por alguno de los siguientes elementos de cálculo más pequeños (llamados *átomos*):
  1. Un átomo de la forma  $R(t_i)$ , donde R es el nombre de una relación y  $t_i$  es una variable de tupla. Esta fórmula identifica el ámbito de la variable tupla  $t_i$  como la relación cuyo nombre es R.
  2. Un átomo de la forma  $t_i . A \text{ op } t_j . B$ , donde op es uno de los operadores de comparación del conjunto  $\{=, <, \leq, >, \geq, \neq\}$ ,  $t_i$  y  $t_j$  son variables de la tupla, y A y B son atributos de las relaciones  $t_i$  y  $t_j$  a las que, respectivamente, engloban.
  3. Un átomo de la forma anterior pero con uno de los operandos siendo una constante y no una variable.

# Cálculo Relacional de Tupla:

## Expresiones y Fórmulas

- ♦ *Cada uno de los átomos anteriores se evalúa como VERDADERO o FALSO para una combinación específica de tuplas (veracidad del átomo).*
- ♦ Mediante estas fórmulas compuestas por átomos queremos saber si los datos almacenados en las BDs cumplen un cierta condición o no, para recuperar esa información.
- ♦ Así por ejemplo para los átomos de la forma  $R(t)$ , si  $t$  está asignada a una tupla que es miembro de la relación  $R$  especificada, ese átomo se evalúa como VERDADERO, en cualquier otro caso FALSO.
  - Ej: EMPLEADO( $t$ ), si  $t$  es una tupla que pertenece a empleado este átomo se evalúa como VERDADERO.
- ♦ Los átomos de tipo 2 o 3, si las variables de tupla están asignadas a tuplas en las que los valores de los atributos especificados de las mismas satisfacen la condición entonces las fórmulas se evalúan como verdaderas.
- ♦ En general una fórmula (COND) está compuesta por uno o varios átomos conectados mediante operadores lógicos, AND, OR y NOT.

# Cálculo Relacional de Tupla:

## Expresiones y Fórmulas

Las relaciones entre fórmulas y átomos cumple las siguientes reglas:

- ♦ **Regla1:** Cada átomo es una fórmula.
- ♦ **Regla2:** Si F1 y F2 son fórmulas entonces también lo son (F1 AND F2), (F1 OR F2), NOT(F1) y NOT(F2). Los valores de veracidad para estas fórmulas se derivan de los obtenidos para F1 y F2 de la siguiente forma:
  - (F1 AND F2) es VERDADERO si F1 y F2 lo son. En cualquier otro caso es FALSO.
  - (F1 OR F2) es FALSO si F1 y F2 lo son. En cualquier otro caso es VERDADERO.
  - NOT(F1) es VERDADERO si F1 es FALSO. En cualquier otro caso es FALSO.
  - NOT(F2) es VERDADERO si F2 es FALSO. En cualquier otro caso es FALSO.

# Cálculo Relacional de Tupla:

## Cuantificadores Existencial y Universal

- ◆ Existen otros símbolos especiales llamados *cuantificadores* que pueden aparecer en la fórmulas: El universal ( $\forall$ ) y el existencial ( $\exists$ ).
- ◆ Conceptos de variables de *tupla libre* y *tupla acotada* de una fórmula: una variable de tupla  $t$  es **ligada o acotada** si está cuantificada (aparece en una clausula  $\exists t$  o  $\forall t$ ), en cualquier otro caso es libre. Las reglas que determinan si es libre o acotada:
  - Una variable de tupla en una fórmula  $F$  que es un átomo es libre en  $F$ .
  - Una variable de tupla  $t$  es libre o acotada en una fórmula construida mediante conexiones lógicas  $\{(F1 \text{ AND } F2), (F1 \text{ OR } F2), \text{NOT}(F1) \text{ y } \text{NOT}(F2)\}$  dependiendo de su estado en  $F1$  o  $F2$ . (**Nota:** En la dos primeras si  $F1$  es libre y  $F2$  acotada, o viceversa, el resultado del AND es acotado y el resultado del OR es libre).
  - Todas las ocurrencias libres de una variable de tupla  $t$  en  $F$  son acotadas en una fórmula  $F'$  de la forma:  $F'=(\exists t)F$  o  $F'=(\forall t)F$ . La variable de tupla es acotada al cuantificador especificado en  $F'$  (notar que los cuantificadores anteriores solo hacen referencia a  $t$ ).
    - $F1: d.\text{NombreDpto}=\text{'investigacion'}$
    - $F2: (\exists t)(d.\text{NumeroDpto}=t.\text{Dno})$
    - $F3: (\forall d)(d.\text{DniDirector}=\text{'333445555'})$

La variable de tupla  $d$  es libre en  $F1$  y  $F2$ , mientras que es acotada en  $F3$  (por el cuantificador  $\forall$ ). La variable  $t$  es acotada respecto al cuantificador ( $\exists$ ) en  $F2$ .

# Cálculo Relacional de Tupla:

## Cuantificadores Existencial y Universal

- ♦ Los valores de comprobación de las fórmulas con estos cuantificadores están descritos por las siguientes reglas:
  - **Regla3:** Si  $F$  es una fórmula, entonces  $F' = (\exists t)F$  también lo es, donde  $t$  es una variable de tupla.  $F'$  es VERDADERO si  $F$  se evalúa como tal *en alguna (al menos una)* tupla asignada a las ocurrencias libres de  $t$  en  $F$ . En cualquier otro caso  $F'$  es FALSO.
  - **Regla4:** Si  $F$  es una fórmula, entonces  $F' = (\forall t)F$  también lo es, donde  $t$  es una variable de tupla.  $F'$  es VERDADERO si  $F$  se evalúa como tal para *cada tupla* asignada a las ocurrencias libres de  $t$  en  $F$ . En cualquier otro caso  $F'$  es FALSO.
- ♦  $\exists$  se dice que es un cuantificador existencial porque una fórmula  $(\exists t)F$  es VERDADERO si existe alguna tupla que haga que  $F$  sea VERDADERO.
- ♦ Para el cuantificador universal,  $(\forall t)F$  es VERDADERO si cada posible tupla que puede asignarse a las ocurrencias libres de  $t$  en  $F$  es sustituida por  $t$ , y  $F$  es VERDADERO para cada una de estas sustituciones. Recibe el nombre de universal (o para todos los cuantificadores) porque cada tupla del universo de tuplas debe hacer que  $F$  sea VERDADERO para la fórmula cuantificada también lo sea.



# Cálculo Relacional de Tupla:

## Operaciones Básicas-Repaso

- ♦ Forma General:  $\{ t \mid F(t) \}$  conjunto de tuplas  $t$  tal que  $F(t)$  es verdadero
- ♦ Relación de intervalo (rango de la tabla A):  $\{ t \mid A(t) \}$  conjunto de tuplas que pertenecen a la relación A
- ♦ Proyección:  $\{ t.A_1, t.A_2 \mid A(t) \}$  extraer los atributos de una tabla
- ♦ Selección:  $\{ t \mid A(t) \text{ and } t.A_1 > 2000 \}$  seccionar tuplas de una tabla
- ♦ Unión:  $\{ t \mid A(t) \text{ or } B(t) \}$  (es necesario la compatibilidad de relaciones)
- ♦ Intersección:  $\{ t \mid A(t) \text{ and } B(t) \}$  (es necesario la compatibilidad de relaciones)
- ♦ Resta:  $\{ t \mid A(t) \text{ and not } B(t) \}$  (es necesario la compatibilidad de relaciones)
- ♦ Producto cartesiano:  $\{ t, s \mid A(t) \text{ and } B(s) \}$
- ♦ Join:  $\{ t, s \mid A(t) \text{ and } B(s) \text{ and } t.A_1 < s.B_1 \}$
- ♦ Join:  $\{ t \mid A(t) \text{ and } ((\exists s)(B(s) \text{ and } t.A_1 < s.B_1)) \}$
- ♦ Equi-Join:  $\{ t, s \mid A(t) \text{ and } B(s) \text{ and } t.A_1 = s.B_1 \}$
- ♦ Equi-Join:  $\{ t \mid A(t) \text{ and } ((\exists s)(B(s) \text{ and } t.A_1 = s.B_1)) \}$

# BD Empresa

## EMPLEADO

▲	Nombre text	Apellido1 text	Apellido2 text	Dni integer	FechaNac date	Direccion text	Sexo "char"	Sueldo numeric	SuperDni integer	Dno integer
1	Jose	Perez	Perez	123456789	1965-09-01	Eloy I, 98	H	30000	333445555	5
2	Alberto	Campos	Sastre	333445555	1955-12-08	Avda Rios, 9	H	40000	888665555	5
3	Alicia	Jimenez	Celaya	999887777	1968-05-12	Gran Via, 38	M	25000	987654321	4
4	Juana	Sainz	Oreja	987654321	1941-06-20	Cerquillas, 67	M	43000	888665555	4
5	Eduardo	Ochoa	Paredes	888665555	1937-11-10	Las Peñas, 1	H	55000	[null]	1
6	Fernan...	Ojeda	Ordoñez	666884444	1962-09-15	Portillo, S/N	M	38000	333445555	5
7	Luis	Pajares	Morera	987987987	1969-03-29	Enebros, 90	H	25000	987654321	4
8	Aurora	Oliva	Avezuela	453453453	1972-07-31	Anton, 6	M	25000	333445555	5

## PROYECTO

▲	NombreProyecto text	NumProyecto integer	UbicacionProyecto text	NumDptoProyecto integer
1	PruductoX	1	Valencia	5
2	ProductoY	2	Sevilla	5
3	ProductoZ	3	Madrid	5
4	Computacion	10	Gijon	4
5	Reorganizacion	20	Madrid	1
6	Comunicaciones	30	Gijon	4

## LOCALIZACIONES\_DPTO

▲	NumeroDpto integer	UbicacionDpto text
1	1	Madrid
2	4	Gijon
3	5	Valencia
4	5	Sevilla
5	5	Madrid

## TRABAJA\_EN

▲	DniEmpleado integer	NumProy integer	Horas numeric
1	123456789	1	32.5
2	123456789	2	7.5
3	666884444	3	40.0
4	453453453	1	20.0
5	453453453	2	20.0
6	333445555	2	10.0
7	333445555	10	10.0
8	333445555	3	10.0
9	333445555	20	10.0
10	999887777	30	30.0
11	999887777	10	10.0
12	987987987	10	35.0
13	987987987	30	5.0
14	987654321	30	20.0
15	987654321	20	15.0
16	888665555	20	[null]

## SUBORDINADO

▲	DniEmpleado integer	NombSubordinado text	Sexo "char"	FechaNac date	Relacion text
1	333445555	Alicia	M	1986-04-05	Hija
2	333445555	Teodoro	H	1983-10-25	Hijo
3	333445555	Luisa	M	1958-05-03	Esposa
4	987654321	Alfonso	H	1942-01-28	Esposo
5	123456789	Miguel	H	1988-01-04	Hijo
6	123456789	Alicia	M	1988-12-30	Hija
7	123456789	Elisa	M	1967-05-05	Esposa

## DEPARTAMENTO

▲	NombreDpto text	NumeroDpto integer	DniDirector integer	FechaIngresoDirector date
1	Investigacion	5	333445555	1988-05-22
2	Administracion	4	987654321	1995-01-01
3	Sede Central	1	888665555	1981-06-19

# Campos, Tuplas y Tablas en una BD

## EMPLEADO

Nombre	Apellido1	Apellido2	<u>Dni</u>	FechaNac	Direccion	Sexo	Sueldo	SuperDni	Dno
text	text	text	integer	date	text	"char"	numeric	integer	integer

## DEPARTAMENTO

NombreDpto	<u>NumeroDpto</u>	DniDirector	FechaIngresoDirector
text	integer	integer	date

## LOCALIZACIONES\_DPTO

<u>NumeroDpto</u>	<u>UbicacionDpto</u>
integer	text

## PROYECTO

NombreProyecto	<u>NumProyecto</u>	UbicacionProyecto	NumDptoProyecto
text	integer	text	integer

## TRABAJA\_EN

<u>DniEmpleado</u>	<u>NumProy</u>	Horas
integer	integer	numeric

## SUBORDINADO

<u>DniEmpleado</u>	<u>NombSubordinado</u>	Sexo	FechaNac	Relacion
integer	text	"char"	date	text

# Cálculo Relacional de Tupla: Ejemplos

- ♦ **Consulta1:** Liste el nombre y la dirección de todos los empleados que trabajan para el departamento 'Investigación':

$\{t.\text{Nombre}, t.\text{Apellido1}, t.\text{Dirección} \mid \text{EMPLEADO}(t) \text{ AND } (\exists d)$   
 $(\text{DEPARTAMENTO}(d) \text{ AND}$   
 $d.\text{NombreDpto} = \text{'Investigación'} \text{ AND}$   
 $d.\text{NumeroDpto} = t.\text{Dno}) \}$

- En cálculo relacional la únicas variables de tupla que son libres aparecen a la izquierda de la barra ( $()$ ).
- En este caso,  $t$  es la única variable de tupla libre.
- Esta variable es **acotada** sucesivamente para cada tupla. Esto quiere decir que si la tupla satisface las condiciones de la consulta, esta se recupera del modo que se especifica.
- La variable de tupla  $d$  es acotada al cuatificador existencial.
- Las condiciones  $\text{EMPLEADO}(t)$  y  $\text{DEPARTAMENTO}(d)$  especifican las relaciones de rango para las variables de tupla  $t$  y  $d$ .
- La condición  $d.\text{NombreDpto} = \text{'Investigación'}$  es una condición de selección (operación de selección del Álgebra Relacional).
- La condición  $d.\text{NumeroDpto} = t.\text{Dno}$  es una condición de concatenación (operación básica del Álgebra Relacional).

# Cálculo Relacional de Tupla: Ejemplos

## Equivalente en SQL

- ◆ Consulta equivalente en SQL:

**SELECT** Nombre, Apellido1, Dirección

**FROM** (EMPLEADO **JOIN** DEPARTAMENTO **ON** Dno= NumeroDpto)

**WHERE** NombreDpto='Investigación';

- ◆ También podemos hacerla como:

**SELECT** Nombre, Apellido1, Dirección

**FROM** EMPLEADO, DEPARTAMENTO

**WHERE** NombreDpto='Investigación' **AND** NumeroDpto =Dno;

- ◆ O también de con el “join” explícito:

**SELECT** Nombre, Apellido1, Dirección

**FROM** EMPLEADO **INNER JOIN** DEPARTAMENTO **ON** NumeroDpto =Dno

**WHERE** NombreDpto='Investigación' ;

# BD Empresa

## EMPLEADO

▲	Nombre text	Apellido1 text	Apellido2 text	Dni integer	FechaNac date	Direccion text	Sexo "char"	Sueldo numeric	SuperDni integer	Dno integer
1	Jose	Perez	Perez	123456789	1965-09-01	Eloy I, 98	H	30000	333445555	5
2	Alberto	Campos	Sastre	333445555	1955-12-08	Avda Rios, 9	H	40000	888665555	5
3	Alicia	Jimenez	Celaya	999887777	1968-05-12	Gran Via, 38	M	25000	987654321	4
4	Juana	Sainz	Oreja	987654321	1941-06-20	Cerquillas, 67	M	43000	888665555	4
5	Eduardo	Ochoa	Paredes	888665555	1937-11-10	Las Peñas, 1	H	55000	[null]	1
6	Fernan...	Ojeda	Ordoñez	666884444	1962-09-15	Portillo, S/N	M	38000	333445555	5
7	Luis	Pajares	Morera	987987987	1969-03-29	Enebros, 90	H	25000	987654321	4
8	Aurora	Oliva	Avezuela	453453453	1972-07-31	Anton, 6	M	25000	333445555	5

## TRABAJA\_EN

▲	DniEmpleado integer	NumProy integer	Horas numeric
1	123456789	1	32.5
2	123456789	2	7.5
3	666884444	3	40.0
4	453453453	1	20.0
5	453453453	2	20.0
6	333445555	2	10.0
7	333445555	10	10.0
8	333445555	3	10.0
9	333445555	20	10.0
10	999887777	30	30.0
11	999887777	10	10.0
12	987987987	10	35.0
13	987987987	30	5.0
14	987654321	30	20.0
15	987654321	20	15.0
16	888665555	20	[null]

## PROYECTO

▲	NombreProyecto text	NumProyecto integer	UbicacionProyecto text	NumDptoProyecto integer
1	PruductoX	1	Valencia	5
2	ProductoY	2	Sevilla	5
3	ProductoZ	3	Madrid	5
4	Computacion	10	Gijon	4
5	Reorganizacion	20	Madrid	1
6	Comunicaciones	30	Gijon	4

## LOCALIZACIONES\_DPTO

▲	NumeroDpto integer	UbicacionDpto text
1	1	Madrid
2	4	Gijon
3	5	Valencia
4	5	Sevilla
5	5	Madrid

## SUBORDINADO

▲	DniEmpleado integer	NombSubordinado text	Sexo "char"	FechaNac date	Relacion text
1	333445555	Alicia	M	1986-04-05	Hija
2	333445555	Teodoro	H	1983-10-25	Hijo
3	333445555	Luisa	M	1958-05-03	Esposa
4	987654321	Alfonso	H	1942-01-28	Esposo
5	123456789	Miguel	H	1988-01-04	Hijo
6	123456789	Alicia	M	1988-12-30	Hija
7	123456789	Elisa	M	1967-05-05	Esposa

## DEPARTAMENTO

▲	NombreDpto text	NumeroDpto integer	DniDirector integer	FechaIngresoDirector date
1	Investigacion	5	333445555	1988-05-22
2	Administracion	4	987654321	1995-01-01
3	Sede Central	1	888665555	1981-06-19

# Campos, Tuplas y Tablas en una BD

## EMPLEADO

Nombre	Apellido1	Apellido2	<u>Dni</u>	FechaNac	Direccion	Sexo	Sueldo	SuperDni	Dno
text	text	text	integer	date	text	"char"	numeric	integer	integer

## DEPARTAMENTO

NombreDpto	<u>NumeroDpto</u>	DniDirector	FechaIngresoDirector
text	integer	integer	date

## LOCALIZACIONES\_DPTO

<u>NumeroDpto</u>	<u>UbicacionDpto</u>
integer	text

## PROYECTO

NombreProyecto	<u>NumProyecto</u>	UbicacionProyecto	NumDptoProyecto
text	integer	text	integer

## TRABAJA\_EN

<u>DniEmpleado</u>	<u>NumProy</u>	Horas
integer	integer	numeric

## SUBORDINADO

<u>DniEmpleado</u>	<u>NombSubordinado</u>	Sexo	FechaNac	Relacion
integer	text	"char"	date	text

# Cálculo Relacional de Tupla: Ejemplos

- ♦ **Consulta 2:** Por cada proyecto ubicado en ‘Gijón’, obtenga el número de departamento que lo gestiona, los apellidos, la fecha de nacimiento y la dirección del director del mismo:

$\{p.\text{NumProyecto}, p.\text{NumDptoProyecto}, e.\text{Apellido1}, e.\text{FechaNac}, e.\text{Dirección} \mid \text{PROYECTO}(p) \text{ AND EMPLEADO}(e) \text{ AND } p.\text{UbicacionProyecto} = \text{'Gijón'} \text{ AND } ((\exists d)(\text{DEPARTAMENTO}(d) \text{ AND } p.\text{NumDptoProyecto} = d.\text{NumeroDpto} \text{ AND } d.\text{DniDirector} = e.\text{Dni}))\}$

- Las variables de tuplas  $p$  y  $e$  son libres. La variable  $d$  es acotada al cuantificador existencial.
- La condición de la consulta se evalúa para cada combinación de tuplas asignadas a ‘ $p$ ’ y a ‘ $e$ ’, seleccionando de esas combinaciones posibles las que son acotadas por las fórmulas de la condición.



# Cálculo Relacional de Tupla: Ejemplos

## Equivalente en SQL

- ◆ Consulta equivalente en SQL:

```
SELECT NumProyecto, NumDptoProyecto, Apellido1, Dirección, FechaNac  
FROM PROYECTO, DEPARTAMENTO, EMPLEADO  
WHERE NumDptoProyecto =NumeroDpto AND DniDirector=Dni AND  
UbicacionProyecto='Gijon';
```

# Cálculo Relacional de Tupla: Ejemplos

- ♦ *Ojo* que distintas variables de tupla de una consulta pueden alcanzar la misma relación, por ejemplo la siguiente consulta.
- ♦ **Consulta 3:** Por cada empleado, recuperar el nombre y primer apellido del empleado, y el nombre y primer apellido de su supervisor inmediato.
- ♦ {e.Nombre, e.Apellido1, s.Nombre, s.Apellido1 | EMPLEADO(e) AND EMPLEADO(s) AND e.SuperDni=s.Dni}
- ♦ Recordar el mismo ejemplo en SQL:

```
SELECT E.Nombre, E.Apellido1, S.Nombre, S.Apellido1  
FROM EMPLEADO AS E, EMPLEADO AS S  
WHERE E.SuperDni=S.Dni;
```

- ♦ La tabla E se va a utilizar como tabla para extraer la información de los empleados que son supervisados. La tabla S se utiliza para extraer la información de los empleados supervisores.
- ♦ Darse cuenta que las tablas E y S son copias de la tabla empleado.

# BD Empresa

## EMPLEADO

▲	Nombre text	Apellido1 text	Apellido2 text	Dni integer	FechaNac date	Direccion text	Sexo "char"	Sueldo numeric	SuperDni integer	Dno integer
1	Jose	Perez	Perez	123456789	1965-09-01	Eloy I, 98	H	30000	333445555	5
2	Alberto	Campos	Sastre	333445555	1955-12-08	Avda Rios, 9	H	40000	888665555	5
3	Alicia	Jimenez	Celaya	999887777	1968-05-12	Gran Via, 38	M	25000	987654321	4
4	Juana	Sainz	Oreja	987654321	1941-06-20	Cerquillas, 67	M	43000	888665555	4
5	Eduardo	Ochoa	Paredes	888665555	1937-11-10	Las Peñas, 1	H	55000	[null]	1
6	Fernan...	Ojeda	Ordoñez	666884444	1962-09-15	Portillo, S/N	M	38000	333445555	5
7	Luis	Pajares	Morera	987987987	1969-03-29	Enebros, 90	H	25000	987654321	4
8	Aurora	Oliva	Avezuela	453453453	1972-07-31	Anton, 6	M	25000	333445555	5

## TRABAJA\_EN

▲	DniEmpleado integer	NumProy integer	Horas numeric
1	123456789	1	32.5
2	123456789	2	7.5
3	666884444	3	40.0
4	453453453	1	20.0
5	453453453	2	20.0
6	333445555	2	10.0
7	333445555	10	10.0
8	333445555	3	10.0
9	333445555	20	10.0
10	999887777	30	30.0
11	999887777	10	10.0
12	987987987	10	35.0
13	987987987	30	5.0
14	987654321	30	20.0
15	987654321	20	15.0
16	888665555	20	[null]

## PROYECTO

▲	NombreProyecto text	NumProyecto integer	UbicacionProyecto text	NumDptoProyecto integer
1	PruductoX	1	Valencia	5
2	ProductoY	2	Sevilla	5
3	ProductoZ	3	Madrid	5
4	Computacion	10	Gijon	4
5	Reorganizacion	20	Madrid	1
6	Comunicaciones	30	Gijon	4

## LOCALIZACIONES\_DPTO

▲	NumeroDpto integer	UbicacionDpto text
1	1	Madrid
2	4	Gijon
3	5	Valencia
4	5	Sevilla
5	5	Madrid

## SUBORDINADO

▲	DniEmpleado integer	NombSubordinado text	Sexo "char"	FechaNac date	Relacion text
1	333445555	Alicia	M	1986-04-05	Hija
2	333445555	Teodoro	H	1983-10-25	Hijo
3	333445555	Luisa	M	1958-05-03	Esposa
4	987654321	Alfonso	H	1942-01-28	Esposo
5	123456789	Miguel	H	1988-01-04	Hijo
6	123456789	Alicia	M	1988-12-30	Hija
7	123456789	Elisa	M	1967-05-05	Esposa

## DEPARTAMENTO

▲	NombreDpto text	NumeroDpto integer	DniDirector integer	FechaIngresoDirector date
1	Investigacion	5	333445555	1988-05-22
2	Administracion	4	987654321	1995-01-01
3	Sede Central	1	888665555	1981-06-19

# Campos, Tuplas y Tablas en una BD

## EMPLEADO

Nombre	Apellido1	Apellido2	<u>Dni</u>	FechaNac	Direccion	Sexo	Sueldo	SuperDni	Dno
text	text	text	integer	date	text	"char"	numeric	integer	integer

## DEPARTAMENTO

NombreDpto	<u>NumeroDpto</u>	DniDirector	FechaIngresoDirector
text	integer	integer	date

## LOCALIZACIONES\_DPTO

<u>NumeroDpto</u>	<u>UbicacionDpto</u>
integer	text

## PROYECTO

NombreProyecto	<u>NumProyecto</u>	UbicacionProyecto	NumDptoProyecto
text	integer	text	integer

## TRABAJA\_EN

<u>DniEmpleado</u>	<u>NumProy</u>	Horas
integer	integer	numeric

## SUBORDINADO

<u>DniEmpleado</u>	<u>NombSubordinado</u>	Sexo	FechaNac	Relacion
integer	text	"char"	date	text

# Cálculo Relacional de Tupla: Ejemplos

- ♦ **Consulta 4:** Enumere el nombre de todos los empleados que trabajan en algún proyecto controlado por el departamento 5:

$\{e.\text{Apellido1}, e.\text{Nombre} \mid \text{EMPLEADO}(e) \text{ AND } ((\exists x) (\exists w) (\text{PROYECTO}(x) \text{ AND } \text{TRABAJA\_EN}(w) \text{ AND } x.\text{NumDptoProyecto}=5 \text{ AND } w.\text{DniEmpleado}=e.\text{Dni} \text{ AND } x.\text{NumProyecto}=w.\text{NumProy})))\}$

- ♦ En SQL:

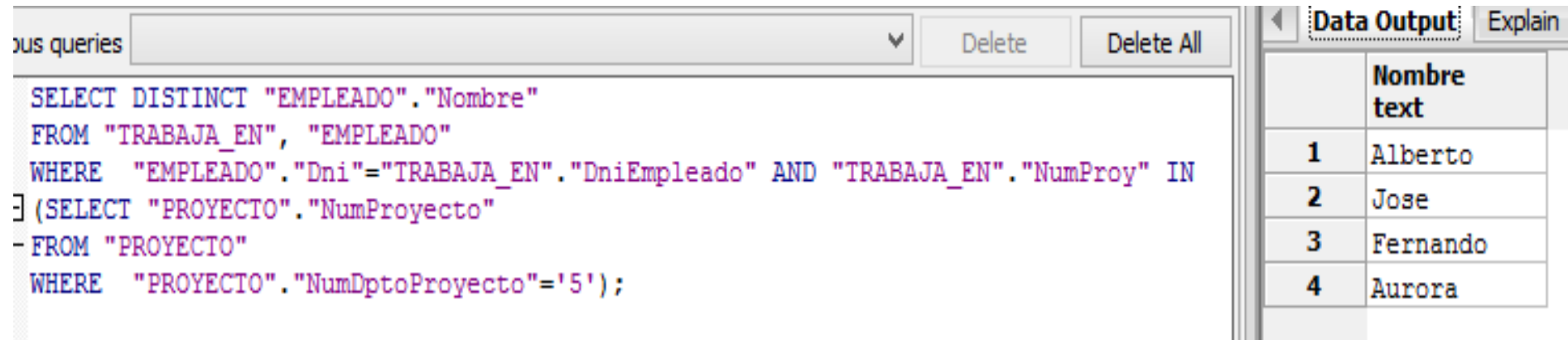
```
SELECT DISTINCT "EMPLEADO"."Nombre"  
FROM "TRABAJA_EN", "EMPLEADO"  
WHERE "EMPLEADO"."Dni"="TRABAJA_EN"."DniEmpleado" AND  
      "TRABAJA_EN"."NumProy" IN  
      (SELECT "PROYECTO"."NumProyecto"  
       FROM "PROYECTO"  
       WHERE "PROYECTO"."NumDptoProyecto"='5');
```

Es una consulta anidada. El segundo SELECT me proporciona los números de proyectos que controla el departamento 5. Estos número se les pasa a la primera consulta.

*¿PODEMOS QUITAR EN ESTE CASO EL ANIDAMIENTO?*

# Cálculo Relacional de Tupla: Ejemplos

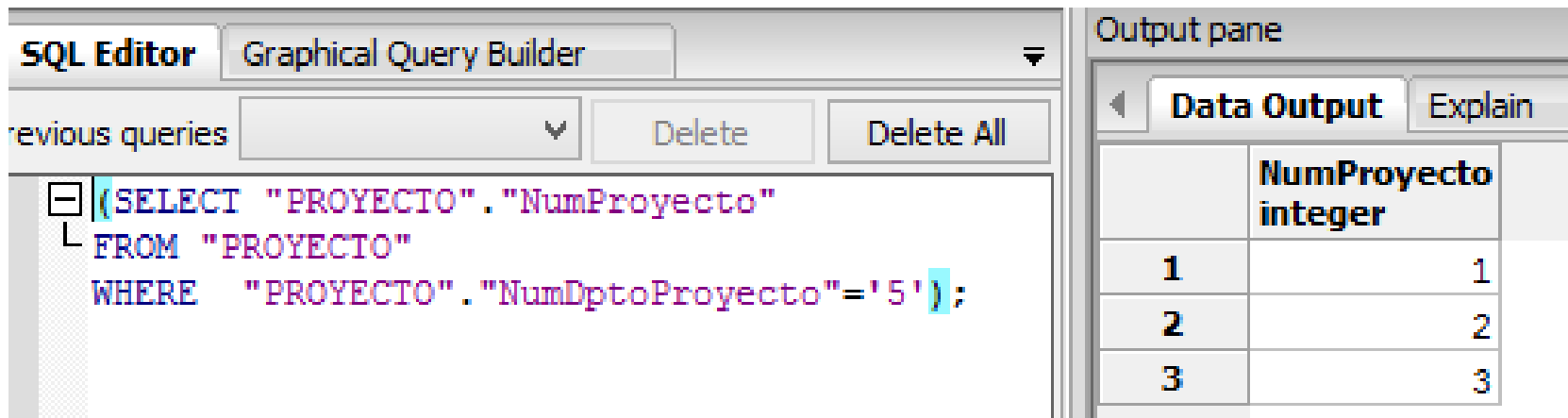
**RECORDTORIO:** Enumere el nombre de todos los empleados que trabajan en algún proyecto controlado por el departamento 5.



```
SELECT DISTINCT "EMPLEADO"."Nombre"
FROM "TRABAJA_EN", "EMPLEADO"
WHERE "EMPLEADO"."Dni"="TRABAJA_EN"."DniEmpleado" AND "TRABAJA_EN"."NumProy" IN
(SELECT "PROYECTO"."NumProyecto"
FROM "PROYECTO"
WHERE "PROYECTO"."NumDptoProyecto"='5');
```

	Nombre text
1	Alberto
2	Jose
3	Fernando
4	Aurora

El segundo SELECT me proporciona los números de proyecto que controla el departamento 5.



```
(SELECT "PROYECTO"."NumProyecto"
FROM "PROYECTO"
WHERE "PROYECTO"."NumDptoProyecto"='5');
```

	NumProyecto integer
1	1
2	2
3	3

# Cálculo Relacional de Tupla: Ejemplos

- ♦ **Consulta 4:** Enumere el nombre de todos los empleados que trabajan en algún proyecto controlado por el departamento 5:

$\{e.\text{Apellido1}, e.\text{Nombre} \mid \text{EMPLEADO}(e) \text{ AND } ((\exists x) (\exists w) (\text{PROYECTO}(x) \text{ AND } \text{TRABAJA\_EN}(w) \text{ AND } x.\text{NumDptoProyecto}=5 \text{ AND } w.\text{DniEmpleado}=e.\text{Dni} \text{ AND } x.\text{NumProyecto}=w.\text{NumProy}))) \}$

- ♦ *En SQL pero sin SELECT anidado:*

```
SELECT DISTINCT "EMPLEADO"."Nombre"  
FROM "TRABAJA_EN", "EMPLEADO", "PROYECTO"  
WHERE "EMPLEADO"."Dni"="TRABAJA_EN"."DniEmpleado" AND  
      "TRABAJA_EN"."NumProy"="PROYECTO"."NumProyecto" AND  
      "PROYECTO"."NumDptoProyecto"='5'
```

# BD Empresa

## EMPLEADO

▲	Nombre text	Apellido1 text	Apellido2 text	Dni integer	FechaNac date	Direccion text	Sexo "char"	Sueldo numeric	SuperDni integer	Dno integer
1	Jose	Perez	Perez	123456789	1965-09-01	Eloy I, 98	H	30000	333445555	5
2	Alberto	Campos	Sastre	333445555	1955-12-08	Avda Rios, 9	H	40000	888665555	5
3	Alicia	Jimenez	Celaya	999887777	1968-05-12	Gran Via, 38	M	25000	987654321	4
4	Juana	Sainz	Oreja	987654321	1941-06-20	Cerquillas, 67	M	43000	888665555	4
5	Eduardo	Ochoa	Paredes	888665555	1937-11-10	Las Peñas, 1	H	55000	[null]	1
6	Fernan...	Ojeda	Ordoñez	666884444	1962-09-15	Portillo, S/N	M	38000	333445555	5
7	Luis	Pajares	Morera	987987987	1969-03-29	Enebros, 90	H	25000	987654321	4
8	Aurora	Oliva	Avezuela	453453453	1972-07-31	Anton, 6	M	25000	333445555	5

## PROYECTO

▲	NombreProyecto text	NumProyecto integer	UbicacionProyecto text	NumDptoProyecto integer
1	PruductoX	1	Valencia	5
2	ProductoY	2	Sevilla	5
3	ProductoZ	3	Madrid	5
4	Computacion	10	Gijon	4
5	Reorganizacion	20	Madrid	1
6	Comunicaciones	30	Gijon	4

## LOCALIZACIONES\_DPTO

▲	NumeroDpto integer	UbicacionDpto text
1	1	Madrid
2	4	Gijon
3	5	Valencia
4	5	Sevilla
5	5	Madrid

## TRABAJA\_EN

▲	DniEmpleado integer	NumProy integer	Horas numeric
1	123456789	1	32.5
2	123456789	2	7.5
3	666884444	3	40.0
4	453453453	1	20.0
5	453453453	2	20.0
6	333445555	2	10.0
7	333445555	10	10.0
8	333445555	3	10.0
9	333445555	20	10.0
10	999887777	30	30.0
11	999887777	10	10.0
12	987987987	10	35.0
13	987987987	30	5.0
14	987654321	30	20.0
15	987654321	20	15.0
16	888665555	20	[null]

## SUBORDINADO

▲	DniEmpleado integer	NombSubordinado text	Sexo "char"	FechaNac date	Relacion text
1	333445555	Alicia	M	1986-04-05	Hija
2	333445555	Teodoro	H	1983-10-25	Hijo
3	333445555	Luisa	M	1958-05-03	Esposa
4	987654321	Alfonso	H	1942-01-28	Esposo
5	123456789	Miguel	H	1988-01-04	Hijo
6	123456789	Alicia	M	1988-12-30	Hija
7	123456789	Elisa	M	1967-05-05	Esposa

## DEPARTAMENTO

▲	NombreDpto text	NumeroDpto integer	DniDirector integer	FechaIngresoDirector date
1	Investigacion	5	333445555	1988-05-22
2	Administracion	4	987654321	1995-01-01
3	Sede Central	1	888665555	1981-06-19



# Campos, Tuplas y Tablas en una BD

## EMPLEADO

Nombre	Apellido1	Apellido2	<u>Dni</u>	FechaNac	Direccion	Sexo	Sueldo	SuperDni	Dno
text	text	text	integer	date	text	"char"	numeric	integer	integer

## DEPARTAMENTO

NombreDpto	<u>NumeroDpto</u>	DniDirector	FechaIngresoDirector
text	integer	integer	date

## LOCALIZACIONES\_DPTO

<u>NumeroDpto</u>	<u>UbicacionDpto</u>
integer	text

## PROYECTO

NombreProyecto	<u>NumProyecto</u>	UbicacionProyecto	NumDptoProyecto
text	integer	text	integer

## TRABAJA\_EN

<u>DniEmpleado</u>	<u>NumProy</u>	Horas
integer	integer	numeric

## SUBORDINADO

<u>DniEmpleado</u>	<u>NombSubordinado</u>	Sexo	FechaNac	Relacion
integer	text	"char"	date	text

# Cálculo Relacional de Tupla: Ejemplos

- ♦ **Consulta 5:** Obtener una lista de los números de los proyectos que impliquen a cualquier empleado cuyo primer apellido sea 'Campos', independientemente de que sean trabajadores o directores del departamento que gestiona dicho proyecto:

*Trabajadores*

$\{p.\text{NumProyecto} \mid \text{PROYECTO}(p) \text{ AND } (((\exists e) (\exists w) \text{ EMPLEADO}(e) \text{ AND TRABAJA\_EN}(w) \text{ w.NumProy}=p.\text{NumProyecto} \text{ AND } e.\text{Apellido1}=\text{'Campos'} \text{ AND } e.\text{Dni}=w.\text{DniEmpleado}))$

OR

$((\exists m) (\exists d) \text{ EMPLEADO}(m) \text{ AND DEPARTAMENTO}(d) \text{ AND } p.\text{NumDptoProyecto}=d.\text{NumeroDpto} \text{ AND } d.\text{DniDirector}=m.\text{Dni} \text{ AND } m.\text{Apellido1}=\text{'Campos'})) \}$

*Directores*

# Cálculo Relacional de Tupla: Ejemplos

- ♦ En SQL la Consulta 5: Obtener una lista de los números de los proyectos que impliquen a cualquier empleado cuyo primer apellido sea 'Campos', independientemente de que sean trabajadores o directores del departamento que gestiona dicho proyecto:

```
SELECT "PROYECTO"."NumProyecto"  
FROM "PROYECTO"  
WHERE "PROYECTO"."NumProyecto" IN  
(SELECT "PROYECTO"."NumProyecto"  
FROM "PROYECTO", "DEPARTAMENTO", "EMPLEADO"  
WHERE "PROYECTO"."NumDptoProyecto"="DEPARTAMENTO"."NumeroDpto" AND  
"DEPARTAMENTO"."DniDirector"="EMPLEADO"."Dni" AND "Apellido1"='Campos')  
OR  
"PROYECTO"."NumProyecto" IN  
(SELECT "TRABAJA_EN"."NumProy"  
FROM "TRABAJA_EN", "EMPLEADO"  
WHERE "TRABAJA_EN"."DniEmpleado"="EMPLEADO"."Dni" AND  
"Apellido1"='Campos');
```

# Cálculo Relacional de Tupla: Ejemplos

**RECORDATORIO:** Obtener una lista de los números de los proyectos que impliquen a cualquier empleado cuyo primer apellido sea 'Campos', independientemente de que sean trabajadores o directores del departamento que gestiona dicho proyecto:

The screenshot shows an SQL Editor window with the following query:

```
SELECT "PROYECTO"."NumProyecto"
FROM "PROYECTO"
WHERE
  "PROYECTO"."NumProyecto" IN
  (SELECT "PROYECTO"."NumProyecto"
   FROM "PROYECTO", "DEPARTAMENTO", "EMPLEADO"
   WHERE "PROYECTO"."NumDptoProyecto"="DEPARTAMENTO"."NumeroDpto"
   AND "DEPARTAMENTO"."DniDirector"="EMPLEADO"."Dni"
   AND "Apellido1"='Campos')
  OR
  "PROYECTO"."NumProyecto" IN
  (SELECT "TRABAJA_EN"."NumProy"
   FROM "TRABAJA_EN", "EMPLEADO"
   WHERE "TRABAJA_EN"."DniEmpleado"="EMPLEADO"."Dni" AND "Apellido1"='Campos');
```

The query is annotated with red circles and arrows:

- A red circle around the first subquery is labeled *Directores* with an arrow pointing to a table of project numbers for directors.
- A red circle around the second subquery is labeled *Empleados* with an arrow pointing to a table of project numbers for employees.

**Directores Table:**

	NumProyecto integer
1	1
2	2
3	3

**Empleados Table:**

	NumProy integer
1	2
2	10
3	3
4	20

**Output pane:**

	NumProyecto integer
1	1
2	2
3	3
4	10
5	20

# Expresiones no seguras

- ♦ Expresiones no seguras
  - Devuelven infinitas tuplas:  
Ejemplo:  $\{ t \mid \text{NOT EMPLEADO}(t) \}$
  - Recupera todas la tuplas del universo que no son tuplas empleado.
  - Solución: evitarlas!
  - La caracterización de consultas seguras y no seguras es compleja –no profundizaremos en ello en este curso.
  - Las equivalencias entre los diferentes formalismos (cálculo, álgebra, cálculo de tuplas vs. de dominio) se dan con salvedad de las expresiones no seguras.

# Álgebra Relacional

- ♦ Se considera como una parte del modelo de datos relacional.
- ♦ Las consultas se hacen mediante operaciones de manera procedural (al contrario del CR que es declarativo como ya hemos visto).
- ♦ Al igual que en CR una consulta en AR da como resultado el conjunto de tuplas que cumplen ciertas condiciones que se desean especificadas por la consulta.
- ♦ Existen varios tipos de operaciones:
  - **Operaciones propias:** desarrolladas propiamente para las bases de datos relacionales (Unarias o Binarias):
    - Selección, Proyección, Renombrado, Concatenación o Combinación.
  - **Operaciones entre conjuntos:**
    - Unión, Intersección, Diferencia de conjuntos, Producto Cartesiano
  - **Operaciones extendidas:** operaciones que se necesitan además de las anteriores:
    - Funciones Agregadas y Concatenación y Unión adicionales.

# Álgebra Relacional: Operaciones Unarias-SELECT

- ♦ La operación unaria SELECCIÓN se emplea para seccionar un conjunto de tuplas de una relación que satisface una condición de selección.
- ♦ Puede ser visto como una **partición horizontal** de la relación: solo la tuplas que cumplen la condición son seleccionadas, el resto se descarta.
- ♦ Ejemplos:
  - $\sigma_{Dno=4}$  (EMPLEADO): selecciona las tuplas de empleado cuyo departamento es 4.
  - $\sigma_{Suelto>30000}$  (EMPLEADO): selecciona las tuplas de empleado cuyo salario es mayor de 30000 €.
- ♦ En general:  $\sigma_{\langle \text{condición de selección} \rangle} (R)$
- ♦ El símbolo  $\sigma$  representa la operación de selección propiamente dicha.
- ♦ La condición de selección es una expresión lógica (o booleana) especificada sobre los atributos de la relación R.

# Álgebra Relacional: Operaciones

## Unarias-SELECT

- ♦ **R en general puede ser una operación de AR**, cuyo resultado es una relación (una tabla o conjunto de tuplas).
- ♦ Obviamente **la R más sencilla es el nombre de una relación** de la BDs, como hemos visto en los ejemplos anteriores.
- ♦ El resultado de la operación selección tiene los mismos atributos de la relación R.
- ♦ La condición de selección puede estar compuesta por las diferentes clausulas:
  - $A \text{ op } B$ , o  $A \text{ op } c$ , donde  $A$  y  $B$  son atributos de una relación ,  $c$  una constante y  $op$  es un operador comparación  $\{=, <, \leq, >, \geq, \neq\}$
- ♦ Estas clausulas pueden estar conectadas por operadores lógicos AND, OR y NOT.
- ♦ **NOTA:** Las condiciones son iguales que las de CR pero no se aplican a variables de tupla, no se utilizan condiciones de esquema indicando la relación de rango de la variable, y no se utilizan los cuantificadores existencial y universal.



# Álgebra Relacional: Operaciones Unarias-SELECT

- ◆ Ejemplos:  $\sigma_{(Dno=4 \text{ AND } Sueldo>25000) \text{ OR } (Dno=5 \text{ AND } Sueldo>30000)}$  (EMPLEADO): secciona las tuplas de la relación empleado que trabajan en el departamento 4 y ganan más de 25000€, o los que trabajan en el departamento 5 y ganan más de 30000.

Nombre	Apellido1	Apellido2	Dni	FechaNac	Dirección	Sexo	Sueldo	SuperDni	Dno
Alberto	Campos	Sastre	333445555	08-12-1955	Avda. Ríos, 9	H	40000	888665555	5
Juana	Sainz	Oreja	987654321	20-06-1941	Cerquillas, 67	M	43000	888665555	4
Fernando	Ojeda	Ordóñez	666884444	15-09-1962	Portillo, s/n	H	38000	333445555	5

# Álgebra Relacional: Operaciones Unarias-SELECT

- ♦ Toda operación de selección cumple:
  - Los atributos de la selección son los mismos que los de la relación.
  - El número de tuplas resultante es siempre menor o igual que el número de tuplas de la relación.
  - La operación selección es conmutativa:
    - $\sigma_{condición1} (\sigma_{condición2} (R)) = \sigma_{condición2} (\sigma_{condición1} (R)) = \sigma_{condición1 \text{ and } condición2} (R)$
    - Esto se puede aplicar a una cascada de condiciones
- ♦ En general la operación selección es una elección de filas de las relaciones (partición horizontal), al contrario de la operación que vamos a estudiar a continuación de **PROYECCIÓN**.
- ♦ Está segunda operación unaria selecciona ciertas columnas de una tabla.
- ♦ Es por que la operación PROYECCIÓN se suele ver como una **partición vertical** de la tabla sobre la que opera.

# Álgebra Relacional: Operaciones Unarias-PROJECT

- ◆ Esta operación selecciona ciertas columnas de una tabla.
- ◆ Es por que la operación PROYECCIÓN se suele ver como una partición vertical de la tabla sobre la que opera.
- ◆ Ejemplo:
  - $\pi_{\text{Apellido1, Nombre, Sueldo}}$  (EMPLEADO): selecciona los atributos Apellido1, Nombre, Sueldo de la tabla EMPLEADO.

Apellido1	Nombre	Sueldo
Pérez	José	30000
Campos	Alberto	40000
Jiménez	Alicia	25000
Sainz	Juana	43000
Ojeda	Fernando	38000
Oliva	Aurora	25000
Pajares	Luis	25000
Ochoa	Eduardo	55000

# Álgebra Relacional: Operaciones Unarias-PROJECT

- ♦ En general:  $\pi_{\langle \text{lista de atributos} \rangle}(\mathbf{R})$
- ♦ El símbolo  $\pi$  representa la operación de proyección propiamente dicha.
- ♦ La lista de atributos contiene a la lista de campos de la relación que queremos extraer de la relación  $R$ .
- ♦ El orden de los atributos en la nueva relación es el orden especificado en la proyección.
- ♦ Si la lista de atributos no incluye ninguna clave candidata de  $R$ , entonces es posible que se obtuviesen tuplas duplicadas.
- ♦ **La operación proyección elimina cualquier tupla que este duplicada, por lo tanto se obtiene una relación válida en el modelo relacional.**
- ♦ Esto es lo que se conoce como **eliminación de duplicados**:
  - $\pi_{\text{Sexo, Sueldo}}(\text{EMPLEADO})$ : aunque formalmente existirían tuplas duplicadas ( $\langle \text{'M'}$ ,  $25000 \rangle$ ), estas se eliminan del resultado.

# Álgebra Relacional: Operaciones Unarias-PROJECT

Sexo	Sueldo
H	30000
H	40000
M	25000
M	43000
H	38000
H	25000
H	55000

- ♦ La eliminación de duplicados lleva implícitamente un proceso de ordenación para detectar esos duplicados y eliminarlos (esto no sucede en SQL).
- ♦ En una operación de proyección siempre se cumple:
  - El número de tuplas resultante es siempre menor o igual que el número de tuplas de la relación (eliminación de duplicados). Es igual estrictamente cuando la lista tiene una clave candidata de R.
  - Propiedad de listas incluidas:  $\pi_{\text{lista1}}(\pi_{\text{lista2}}(R)) = \pi_{\text{lista1}}(R)$ , siempre y cuando  $\text{lista2} \subset \text{lista1}$ , si esto no pasa la expresión es incorrecta.
  - La operación proyección **NO es conmutativa**.

# Álgebra Relacional: Operaciones Unarias-RENAME

- ♦ Al resultado de las operaciones anteriores no se les asigna ningún nombre.
- ♦ Se pueden realizar todas las operaciones de AR una tras otra, anidando dichas operaciones (sin tener que asignar nombres intermedios).
- ♦ Podemos querer crear **relaciones intermedias** y asignar nombre a esas relaciones intermedias.
- ♦ Ejemplo: **Recuperar el nombre, el primer apellido y el sueldo de todos los empleados que trabajan en el departamento 5**, para ello debemos aplicar una selección y una proyección:
  - $\pi_{\text{Nombre, Apellido1, Sueldo}}(\sigma_{\text{Dno}=5}(\text{EMPLEADO}))$ : Mostrar la figura (a) siguiente.
  - Podemos mostrar la secuencia de operaciones dando un nombre a cada una de ellas:
    - $\text{DEP5\_EMPS} \leftarrow \sigma_{\text{Dno}=5}(\text{EMPLEADO})$
    - $\text{RESULTADO} \leftarrow \pi_{\text{Nombre, Apellido1, Sueldo}}(\text{DEP5\_EMPS})$ .
- ♦ También podemos utilizar la **técnica de RENOMBRAR** los atributos en las relaciones intermedias y resultantes.
- ♦ Es muy útil con operaciones más complejas de unión y concatenación.

# Álgebra Relacional: Operaciones Unarias-RENAME

(a)

Nombre	Apellido1	Sueldo
José	Pérez	30000
Alberto	Campos	40000
Fernando	Ojeda	38000
Aurora	Oliva	25000

(b)

**TEMP**

Nombre	Apellido1	Apellido2	Dni	FechaNac	Dirección	Sexo	Sueldo	SuperDni	Dno
José	Pérez	Pérez	123456789	01-09-1965	Eloy I, 98	H	30000	333445555	5
Alberto	Campos	Sastre	333445555	08-12-1955	Avda. Ríos, 9	H	40000	888665555	5
Fernando	Ojeda	Ordóñez	666884444	15-09-1962	Portillo, s/n	H	38000	333445555	5
Aurora	Oliva	Avezuela	453453453	31-07-1972	Antón, 6	M	25000	333445555	5

R

NuevoNombre	NuevoApellido	NuevoSueldo
José	Pérez	30000
Alberto	Campos	40000
Fernando	Ojeda	38000
Aurora	Oliva	25000

# Álgebra Relacional: Operaciones Unarias-RENAME

- ♦ Para renombrar los atributos lo hacemos de la siguiente forma con el ejemplo anterior (dos tipos de renombrado):
  - $TEMP \leftarrow \sigma_{Dno=5} (EMPLEADO)$  (*operación selección*)
  - $R(NuevoNombre, NuevoApellido, NuevoSueldo) \leftarrow \pi_{Nombre, Apellido1, Sueldo} (TEMP)$  (*operación proyección*)Mostrar la figura (b) y (c) anterior página.
- ♦ En la operación selección no se renombran los atributos y por lo tanto tienen el mismo nombre y en mismo orden.
- ♦ En la operación proyección los atributos tienen los nombres renombrados.
- ♦ Así una **operación de renombrado es un operador unario**.



# Álgebra Relacional: Operaciones Unarias-RENAME

- ♦ La forma genérica de una operación de renombrado aplicada a una relación  $R$  de grado  $n$  puede ser:

$$- \rho_{S(B_1, B_2, \dots, B_n)}(R), \rho_S(R), \rho_{(B_1, B_2, \dots, B_n)}(R)$$

- ♦ Donde el símbolo  $\rho$  representa la operación de renombrado.
- ♦  $S$  es el nombre de la nueva relación.
- ♦  $B_1, B_2, \dots, B_n$  son los nuevos atributos y  $R_1, R_2, \dots, R_n$  son los de  $R$ .
- ♦ La primera expresión renombra tanto la relación como los atributos ( $\rho_{S(B_1, B_2, \dots, B_n)}(R)$ ).
- ♦ La segunda expresión solo renombra la relación ( $\rho_S(R)$ ).
- ♦ La tercera solo los atributos ( $\rho_{(B_1, B_2, \dots, B_n)}(R)$ ).
- ♦ Si suponemos que los atributos de  $R$  son  $(A_1, A_2, \dots, A_n)$  por este orden entonces cada  $A_i$  se renombra a  $B_i$ .
- ♦ Para renombrar solo el atributo  $R_i$  de la relación  $R$  por  $B_i$ :  $\rho_{B_i/R_i}(R)$ .

# Álgebra Relacional: Operaciones Entre Conjuntos (Binarias)

- ♦ UNIÓN (UNION)
- ♦ INTERSECCIÓN (INTERSECTION)
- ♦ DIFERENCIA (MINUS)
- ♦ PRODUCTO CARTESIANO o PRODUCTO CUZADO (CROSS JOIN) (necesario para explicar bien la operación extendida de concatenación posteriormente)

# BD Empresa

## EMPLEADO

▲	Nombre text	Apellido1 text	Apellido2 text	Dni integer	FechaNac date	Direccion text	Sexo "char"	Sueldo numeric	SuperDni integer	Dno integer
1	Jose	Perez	Perez	123456789	1965-09-01	Eloy I, 98	H	30000	333445555	5
2	Alberto	Campos	Sastre	333445555	1955-12-08	Avda Rios, 9	H	40000	888665555	5
3	Alicia	Jimenez	Celaya	999887777	1968-05-12	Gran Via, 38	M	25000	987654321	4
4	Juana	Sainz	Oreja	987654321	1941-06-20	Cerquillas, 67	M	43000	888665555	4
5	Eduardo	Ochoa	Paredes	888665555	1937-11-10	Las Peñas, 1	H	55000	[null]	1
6	Fernan...	Ojeda	Ordoñez	666884444	1962-09-15	Portillo, S/N	M	38000	333445555	5
7	Luis	Pajares	Morera	987987987	1969-03-29	Enebros, 90	H	25000	987654321	4
8	Aurora	Oliva	Avezuela	453453453	1972-07-31	Anton, 6	M	25000	333445555	5

## PROYECTO

▲	NombreProyecto text	NumProyecto integer	UbicacionProyecto text	NumDptoProyecto integer
1	PruductoX	1	Valencia	5
2	ProductoY	2	Sevilla	5
3	ProductoZ	3	Madrid	5
4	Computacion	10	Gijon	4
5	Reorganizacion	20	Madrid	1
6	Comunicaciones	30	Gijon	4

## LOCALIZACIONES\_DPTO

▲	NumeroDpto integer	UbicacionDpto text
1	1	Madrid
2	4	Gijon
3	5	Valencia
4	5	Sevilla
5	5	Madrid

## TRABAJA\_EN

▲	DniEmpleado integer	NumProy integer	Horas numeric
1	123456789	1	32.5
2	123456789	2	7.5
3	666884444	3	40.0
4	453453453	1	20.0
5	453453453	2	20.0
6	333445555	2	10.0
7	333445555	10	10.0
8	333445555	3	10.0
9	333445555	20	10.0
10	999887777	30	30.0
11	999887777	10	10.0
12	987987987	10	35.0
13	987987987	30	5.0
14	987654321	30	20.0
15	987654321	20	15.0
16	888665555	20	[null]

## SUBORDINADO

▲	DniEmpleado integer	NombSubordinado text	Sexo "char"	FechaNac date	Relacion text
1	333445555	Alicia	M	1986-04-05	Hija
2	333445555	Teodoro	H	1983-10-25	Hijo
3	333445555	Luisa	M	1958-05-03	Esposa
4	987654321	Alfonso	H	1942-01-28	Esposo
5	123456789	Miguel	H	1988-01-04	Hijo
6	123456789	Alicia	M	1988-12-30	Hija
7	123456789	Elisa	M	1967-05-05	Esposa

## DEPARTAMENTO

▲	NombreDpto text	NumeroDpto integer	DniDirector integer	FechaIngresoDirector date
1	Investigacion	5	333445555	1988-05-22
2	Administracion	4	987654321	1995-01-01
3	Sede Central	1	888665555	1981-06-19

# Campos, Tuplas y Tablas en una BD

## EMPLEADO

Nombre	Apellido1	Apellido2	<u>Dni</u>	FechaNac	Direccion	Sexo	Sueldo	SuperDni	Dno
text	text	text	integer	date	text	"char"	numeric	integer	integer

## DEPARTAMENTO

NombreDpto	<u>NumeroDpto</u>	DniDirector	FechaIngresoDirector
text	integer	integer	date

## LOCALIZACIONES\_DPTO

<u>NumeroDpto</u>	<u>UbicacionDpto</u>
integer	text

## PROYECTO

NombreProyecto	<u>NumProyecto</u>	UbicacionProyecto	NumDptoProyecto
text	integer	text	integer

## TRABAJA\_EN

<u>DniEmpleado</u>	<u>NumProy</u>	Horas
integer	integer	numeric

## SUBORDINADO

<u>DniEmpleado</u>	<u>NombSubordinado</u>	Sexo	FechaNac	Relacion
integer	text	"char"	date	text

# Álgebra Relacional: Operaciones Entre Conjuntos ( $\cup$ , $\cap$ , $-$ )

- ♦ Estas operaciones de AR son las correspondientes a la **operativa matemática sobre conjuntos**.
- ♦ Ejemplo de la operación UNIÓN: Recuperar los Documentos Nacionales de Identidad de todos los empleados que, o bien trabajan en el departamento 5 o bien supervisan a estos:
  1.  $DEP5\_EMPS \leftarrow \sigma_{Dno=5} (EMPLEADO)$
  2.  $RESULTADO1 \leftarrow \pi_{Dni} (DEP5\_EMPS)$
  3.  $RESULTADO2(Dni) \leftarrow \pi_{SuperDni} (DEP5\_EMPS)$
  4.  $RESULTADO \leftarrow RESULTADO1 \cup RESULTADO2$ 
    - RESULTADO1 tiene el Dni de todos los empleados del departamento 5, RESULTADO2 tiene el Dni de todos aquellos empleados que supervisan directamente a los del primer grupo.
    - La unión de ambos es el conjunto de las tuplas que están en RESULTADO1 o están en RESULTADO2 o en ambas.
- ♦ Mostrar siguiente figura.

# Álgebra Relacional: Operaciones Entre Conjuntos ( $\cup$ , $\cap$ , $-$ )

RESULTADO1

Dni
123456789
333445555
666884444
453453453

RESULTADO2

Dni
333445555
888665555

RESULTADO

Dni
123456789
333445555
666884444
453453453
888665555

# Álgebra Relacional: Operaciones Entre Conjuntos ( $\cup$ , $\cap$ , $-$ )

- ♦ Las operaciones entre conjuntos sirven para combinar elementos entre conjuntos como uno desee.
- ♦ Todas las operaciones son binarias, ya que actúan sobre dos conjuntos de tuplas.
- ♦ Concepto de **compatibilidad de tuplas**: dos relaciones  $R(A_1, A_2, \dots, A_n)$  y  $S(B_1, B_2, \dots, B_n)$ , se dice que son de *unión compatible* si ambas relaciones tienen el mismo grado  $n$  y el  $\text{dom}(A_i) = \text{dom}(B_i)$  con  $i$  entre 1 y  $n$ .
- ♦ Las operaciones unión, intersección y diferencia se tienen que aplicar entre conjuntos de tuplas compatibles (**en el producto cartesiano no es necesario**).
- ♦ Suponiendo  $R$  y  $S$  son dos conjuntos de tuplas compatibles las tres operaciones se definen:
  - **UNIÓN:  $R \cup S$**  es una relación que incluye todas la tuplas que están en  $R$  o están en  $S$  o en ambas  $R$  y  $S$  (obviamente no hay duplicados, “o”).
  - **INTERSECCIÓN:  $R \cap S$**  es una relación que incluye todas la tuplas que están en  $R$  y  $S$ .
  - **DIFERENCIA:  $R - S$**  es una relación que incluye todas la tuplas que están en  $R$  pero no están en  $S$ .

# Álgebra Relacional: Operaciones Entre Conjuntos ( $\cup$ , $\cap$ , $-$ )

(a) ESTUDIANTE

Nombre	Apellido
Susana	Gómez
Luis	Campos
Juan	Garrido
Bárbara	Durán
Amanda	González
Joaquín	Martín
Ernesto	Flores

PROFESOR

Nom	Apell
Antonio	Fernández
Ricardo	Adriano
Susana	Gómez
Francisco	Peláez
Luis	Campos

(b)  $E \cup P$

Nombre	Apellido
Susana	Gómez
Luis	Campos
Juan	Garrido
Bárbara	Durán
Amanda	González
Joaquín	Martín
Ernesto	Flores
Antonio	Fernández
Ricardo	Adriano
Francisco	Peláez

(c)  $E \cap P$

Nombre	Apellido
Susana	Gómez
Luis	Campos

(d)  $E - P$

Nombre	Apellido
Juan	Garrido
Bárbara	Durán
Amanda	González
Joaquín	Martín
Ernesto	Flores

(e)  $P - E$

Nom	Apell
Antonio	Fernández
Ricardo	Adriano
Francisco	Peláez



# Álgebra Relacional: Operaciones Entre Conjuntos ( $\cup$ , $\cap$ , $-$ ), Propiedades

- ♦ En todas las siguientes propiedades se suponen dos relaciones compatibles o de unión compatible  $R(A_1, A_2, \dots, A_n)$  y  $S(B_1, B_2, \dots, B_n)$ .
- ♦ La unión e intersección son operaciones conmutativas y asociativas:
  - $R \cup S = S \cup R$  y  $R \cap S = S \cap R$
  - $R \cup (S \cup T) = (R \cup S) \cup T$  y  $R \cap (S \cap T) = (R \cap S) \cap T$
- ♦ La operación menos no es conmutativa:
  - $R - S \neq S - R$  (ver figura anterior)
- ♦ La intersección se puede poner en términos de la unión y diferencia:
  - $R \cap S = R \cup S - (R - S) - (S - R)$
- ♦ Siempre la salida de las operaciones tiene el mismo número de atributos de los conjuntos.
- ♦  $\min(|R|, |S|) \leq |R \cup S| \leq |R| + |S|$
- ♦  $|R \cap S| \leq \min(|R|, |S|)$

# Álgebra Relacional: Operaciones Entre Conjuntos ( $\cup$ , $\cap$ , $-$ ), Propiedades

- ♦  $\sigma_c(R) \cap \sigma_d(S) = \sigma_{c \text{ and } d}(R \cap S)$
- ♦  $\sigma_c(R) \cup \sigma_d(R) = \sigma_{c \text{ or } d}(R)$

# Álgebra Relacional: Operaciones Entre Conjuntos ( $\times$ )

- ♦ **Producto Cartesiano:** Se trata también de una operación de conjuntos binarios, aunque los operandos no tienen porque ser de unión compatible.
- ♦ En general  $R(A_1, A_2, \dots, A_n) \times S(B_1, B_2, \dots, B_m)$  es una relación  $Q$  de grado  $n+m$  atributos  $Q(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$ .
- ♦ En la relación  $Q$  resultante tiene una tupla por cada combinación de éstas (una para  $R$  y otra para  $S$ ).
- ♦ Por tanto si  $R$  tiene  $n_R$  tuplas y  $S$  tiene  $n_S$  tuplas entonces  $R \times S$  tendrá  $n_R * n_S$  tuplas.
- ♦ Supongamos la siguiente secuencia de operaciones:
  1.  $EMPLEADAS\_FEMENINAS \leftarrow \sigma_{\text{Sexo}='M'}(EMPLEADO)$
  2.  $NOMBRE\_EMPLEADOS \leftarrow \pi_{\text{Nombre, Apellido1, Dni}}(EMPLEADAS\_FEMENINAS)$
  3.  $EMPLEADOS\_SUBORDINADOS \leftarrow NOMBRE\_EMPLEADOS \times SUBORDINADO$
  4.  $SUBORDINADOS\_ACTUALES \leftarrow \sigma_{\text{Dni}=\text{DniEmpleado}}(EMPLEADOS\_SUBORDINADOS)$
  5.  $RESULTADO \leftarrow \pi_{\text{Nombre, Apellido1, NombreSubordinado}}(SUBORDINADOS\_ACTUALES)$
- ♦ El producto cartesiano solo tiene sentido si se hace a continuación que combine las relaciones componentes en la manera que deseamos.

# Álgebra Relacional: Operaciones Entre Conjuntos ( $\times$ )

**EMPLEADAS\_FEMENINAS**

Nombre	Apellido1	Apellido2	Dni	FechaNac	Dirección	Sexo	Sueldo	SuperDni	Dno
Alicia	Jiménez	Celaya	999887777	12-05-1968	Gran Vía, 38	M	25000	987654321	4
Juana	Sainz	Oreja	987654321	20-06-1941	Cerquillas, 67	M	43000	888665555	4
Aurora	Oliva	Avezuela	453453453	31-07-1972	Antón, 6	M	25000	333445555	5

**NOMBRES\_EMPLEADOS**

Nombre	Apellido1	Dni
Alicia	Jiménez	999887777
Juana	Sainz	987654321
Aurora	Oliva	453453453

$\times$

**SUBORDINADO**

<u>DniEmpleado</u>	<u>NombSubordinado</u>	Sexo	FechaNac	Relación
333445555	Alicia	M	05-04-1986	Hija
333445555	Teodoro	H	25-10-1983	Hijo
333445555	Luisa	M	03-05-1958	Esposa
987654321	Alfonso	H	28-02-1942	Esposo
123456789	Miguel	H	04-01-1988	Hijo
123456789	Alicia	M	30-12-1988	Hija
123456789	Elisa	M	05-05-1967	Esposa

# Álgebra Relacional: Operaciones Entre Conjuntos ( $\times$ )

EMPLEADOS\_SUBORDINADOS

Nombre	Apellido1	Dni	DniEmpleado	NombreSubordinado	Sexo	FechaNac	...
Alicia	Jiménez	999887777	333445555	Ana	M	05-04-1986	...
Alicia	Jiménez	999887777	333445555	Teodoro	H	25-10-1983	...
Alicia	Jiménez	999887777	333445555	Ruth	M	03-05-1958	...
Alicia	Jiménez	999887777	987654321	Augusto	H	28-02-1942	...
Alicia	Jiménez	999887777	123456789	Miguel	H	01-04-1988	...
Alicia	Jiménez	999887777	123456789	Ana	M	30-12-1988	...
Alicia	Jiménez	999887777	123456789	Elisa	M	05-05-1967	...
Juana	Sainz	987654321	333445555	Ana	M	05-04-1986	...
Juana	Sainz	987654321	333445555	Teodoro	H	25-10-1983	...
Juana	Sainz	987654321	333445555	Ruth	M	03-05-1958	...
Juana	Sainz	987654321	987654321	Augusto	H	28-02-1942	...
Juana	Sainz	987654321	123456789	Miguel	H	04-01-1988	...
Juana	Sainz	987654321	123456789	Ana	M	30-12-1988	...
Juana	Sainz	987654321	123456789	Elisa	M	05-05-1967	...
Aurora	Oliva	453453453	333445555	Ana	M	05-04-1986	...
Aurora	Oliva	453453453	333445555	Teodoro	H	25-10-1983	...
Aurora	Oliva	453453453	333445555	Ruth	M	03-05-1958	...
Aurora	Oliva	453453453	987654321	Augusto	H	28-02-1942	...
Aurora	Oliva	453453453	123456789	Miguel	H	04-01-1988	...
Aurora	Oliva	453453453	123456789	Ana	M	30-12-1988	...
Aurora	Oliva	453453453	123456789	Elisa	M	05-05-1967	...

# Álgebra Relacional: Operaciones Entre Conjuntos ( $\times$ )

SUBORDINADOS\_ACTUALES

Nombre	Apellido1	Dni	DniEmpleado	NombreSubordinado	Sexo	FechaNac	...
Juana	Sainz	987654321	987654321	Augusto	H	28-02-1942	...

RESULTADO

Nombre	Apellido1	NombreSubordinado
Juana	Sainz	Augusto

- ◆ El producto cartesiano crea tuplas de con los atributos combinados de ambas relaciones.
- ◆ Solo podemos hacer una selección de tuplas de las dos relaciones especificando una condición de selección apropiada.
- ◆ Esto es lo que veremos ahora como la operación de **CONCATENACIÓN**.
- ◆ En el ejemplo anterior queremos combinar una tupla de empleada femenina solo con la de sus subordinados particulares (es decir la tuplas de SUBORDINADO cuyos valores de DniEmpleado coincidan con Dni de EMPLEADO).

# Álgebra Relacional: Operaciones Binarias-CONCATENACIÓN

- ♦ El símbolo de la concatenación o JOIN es  $|><|$ , se emplea para combinar tuplas relacionadas de dos relaciones en una sola.
- ♦ Esta operación es fundamental ya que permite procesar relaciones entre relaciones.
- ♦ Supongamos que queremos recuperar el nombre del director de cada departamento, para ello necesitamos combinar la tuplas departamento y empleado a través de la operación concatenación:

1.  $\text{DIRECTOR\_DPTO} \leftarrow \text{DEPARTAMENTO} |><|_{\text{DniDirector}=\text{Dni}} \text{EMPLEADO}$
2.  $\text{RESULTADO} \leftarrow \pi_{\text{NombreDpto}, \text{Apellido1}, \text{Nombre}}(\text{DIRECTOR\_DPTO})$

DIRECTOR\_DPTO

NombreDpto	NúmeroDpto	DniDirector	...	Nombre	Apellido1	Apellido2	Dni	...
Investigación	5	333445555	...	Alberto	Campos	Sastre	333445555	...
Administración	4	987654321	...	Juana	Sainz	Oreja	987654321	...
Sede Central	1	888665555	...	Eduardo	Ochoa	Paredes	888665555	...

# Álgebra Relacional: Operaciones Binarias-CONCATENACIÓN

- ♦ La concatenación, como ya vimos puede ser enunciada como un producto cartesiano más una selección posterior ( $|><| = \times \text{ más } \sigma$ ).
- ♦ Consideremos el ejemplo de hace unas transparencias:
  1. EMPLEADOS\_SUBORDINADOS  $\leftarrow$  NOMBRE\_EMPLEADOS  $\times$  SUBORDINADO
  2. SUBORDINADOS\_ACTUALES  $\leftarrow \sigma_{\text{Dni}=\text{DniEmpleado}}(\text{EMPLEADOS\_SUBORDINADOS})$
- ♦ Estas dos operaciones se pueden sustituir por la operación concatenación:
  - SUBORDINADOS\_ACTUALES  $\leftarrow$  NOMBRES\_EMPLEADOS  $|><|_{\text{Dni}=\text{DniEmpleado}}$  SUBORDINADO
- ♦ La forma general de la concatenación en dos relaciones  $R(A_1, A_2, \dots, A_n)$  y  $S(B_1, B_2, \dots, B_m)$  es:

$$R |><|_{\text{condición de conexión}} S$$

- ♦ El resultado de la concatenación es una relación  $Q$  de  $n+m$  atributos  $Q(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$ , por este orden que tiene una tupla por cada combinación de éstas (una para  $R$  y otra para  $S$ ), siempre que dicha combinación satisfaga la condición de combinación.
- ♦ Aquí radica la **diferencia principal** entre PRODUCTO CARTESIANO y CONCATENACIÓN.



# Álgebra Relacional: Operaciones Binarias-CONCATENACIÓN

- ♦ Una condición general de conexión la podemos expresar como:
  - $\langle \text{condición} \rangle \text{ AND } \langle \text{condición} \rangle \text{ AND } \dots \text{ AND } \langle \text{condición} \rangle$
- ♦ Solo aparecen la tuplas cuyas condiciones de conexión se evalúan como verdaderas.
- ♦ Donde cada  $\langle \text{condición} \rangle$  es de la forma  $A_i \theta B_j$ , con  $A_i$  siendo un atributo de la relación  $R$  y  $B_j$  es un atributo de la relación  $S$  y  $\theta$  un operador de comparación  $\{=, <, \leq, >, \geq, \neq\}$ .
- ♦ Una concatenación con una condición de conexión de este tipo recibe el nombre de ASOCIACIÓN (THETA JOIN).
- ♦ Hay que observar que en una ASOCIACIÓN las tuplas cuyos atributos de conexión son NULL o aquellas cuya condición de conexión se evalúa como falsa no aparecen en el resultado (la concatenación no preserva necesariamente toda la información de las relaciones participantes).

# Álgebra Relacional: CONCATENACIÓN EQUIJOIN y NATURAL

- ◆ El uso más simple de las concatenaciones supone el uso de condiciones de concatenación con solo comparaciones de igualdad: EQUIJOIN
- ◆ En EQUIJOIN siempre tenemos atributos que cuentan con valores idénticos en cada tupla (ver en la figura anterior Dni y DniDirector).
- ◆ Ya que los valores idénticos de los atributos en las tuplas son innecesarios, se creó una nueva operación llamada CONCATENACIÓN NATURAL (NATURAL JOIN), cuyo símbolo es  $|><|$  (o en el libro el símbolo \*).
- ◆ **El natural join se deshace del segundo atributo superfluo en una condición EQUIJOIN (elimina así la redundancia que se genera por un JOIN).**
- ◆ La concatenación equijoin estándar precisa que los dos atributos (o dos combinaciones de atributos idénticas en cada una de las tablas) de conexión tengan el mismo nombre en las dos relaciones.
- ◆ En el siguiente ejemplo primero **renombramos** un atributo **NumeroDpto**  $\leftarrow$  **NumeroDptoProyecto** y luego aplicamos la concatenación natural:
  - $\text{PROYECTO\_DPTO} \leftarrow \text{PROYECTO} |><| \rho_{S(\text{NombreDpto}, \text{NumDptoProyecto}, \text{DniDirector}, \text{FechaIngresoDirector})}(\text{DEPARTAMENTO})$ .

# BD Empresa

## EMPLEADO

▲	Nombre text	Apellido1 text	Apellido2 text	Dni integer	FechaNac date	Direccion text	Sexo "char"	Sueldo numeric	SuperDni integer	Dno integer
1	Jose	Perez	Perez	123456789	1965-09-01	Eloy I, 98	H	30000	333445555	5
2	Alberto	Campos	Sastre	333445555	1955-12-08	Avda Rios, 9	H	40000	888665555	5
3	Alicia	Jimenez	Celaya	999887777	1968-05-12	Gran Via, 38	M	25000	987654321	4
4	Juana	Sainz	Oreja	987654321	1941-06-20	Cerquillas, 67	M	43000	888665555	4
5	Eduardo	Ochoa	Paredes	888665555	1937-11-10	Las Peñas, 1	H	55000	[null]	1
6	Fernan...	Ojeda	Ordoñez	666884444	1962-09-15	Portillo, S/N	M	38000	333445555	5
7	Luis	Pajares	Morera	987987987	1969-03-29	Enebros, 90	H	25000	987654321	4
8	Aurora	Oliva	Avezuela	453453453	1972-07-31	Anton, 6	M	25000	333445555	5

## PROYECTO

▲	NombreProyecto text	NumProyecto integer	UbicacionProyecto text	NumDptoProyecto integer
1	PruductoX	1	Valencia	5
2	ProductoY	2	Sevilla	5
3	ProductoZ	3	Madrid	5
4	Computacion	10	Gijon	4
5	Reorganizacion	20	Madrid	1
6	Comunicaciones	30	Gijon	4

## SUBORDINADO

▲	DniEmpleado integer	NombSubordinado text	Sexo "char"	FechaNac date	Relacion text
1	333445555	Alicia	M	1986-04-05	Hija
2	333445555	Teodoro	H	1983-10-25	Hijo
3	333445555	Luisa	M	1958-05-03	Esposa
4	987654321	Alfonso	H	1942-01-28	Esposo
5	123456789	Miguel	H	1988-01-04	Hijo
6	123456789	Alicia	M	1988-12-30	Hija
7	123456789	Elisa	M	1967-05-05	Esposa

## TRABAJA\_EN

▲	DniEmpleado integer	NumProy integer	Horas numeric
1	123456789	1	32.5
2	123456789	2	7.5
3	666884444	3	40.0
4	453453453	1	20.0
5	453453453	2	20.0
6	333445555	2	10.0
7	333445555	10	10.0
8	333445555	3	10.0
9	333445555	20	10.0
10	999887777	30	30.0
11	999887777	10	10.0
12	987987987	10	35.0
13	987987987	30	5.0
14	987654321	30	20.0
15	987654321	20	15.0
16	888665555	20	[null]

## LOCALIZACIONES\_DPTO

▲	NumeroDpto integer	UbicacionDpto text
1	1	Madrid
2	4	Gijon
3	5	Valencia
4	5	Sevilla
5	5	Madrid

## DEPARTAMENTO

▲	NombreDpto text	NumeroDpto integer	DniDirector integer	FechaIngresoDirector date
1	Investigacion	5	333445555	1988-05-22
2	Administracion	4	987654321	1995-01-01
3	Sede Central	1	888665555	1981-06-19

# Campos, Tuplas y Tablas en una BD

## EMPLEADO

Nombre	Apellido1	Apellido2	<u>Dni</u>	FechaNac	Direccion	Sexo	Sueldo	SuperDni	Dno
text	text	text	integer	date	text	"char"	numeric	integer	integer

## DEPARTAMENTO

NombreDpto	<u>NumeroDpto</u>	DniDirector	FechaIngresoDirector
text	integer	integer	date

## LOCALIZACIONES\_DPTO

<u>NumeroDpto</u>	<u>UbicacionDpto</u>
integer	text

## PROYECTO

NombreProyecto	<u>NumProyecto</u>	UbicacionProyecto	NumDptoProyecto
text	integer	text	integer

## TRABAJA\_EN

<u>DniEmpleado</u>	<u>NumProy</u>	Horas
integer	integer	numeric

## SUBORDINADO

<u>DniEmpleado</u>	<u>NombSubordinado</u>	Sexo	FechaNac	Relacion
integer	text	"char"	date	text

# Álgebra Relacional: CONCATENACIÓN EQUIJOIN y NATURAL

- ◆ Este ejemplo anterior:
  - $\text{PROYECTO\_DPTO} \leftarrow \text{PROYECTO} \bowtie \rho_{S(\text{NombreDpto}, \text{NumDptoProyecto}, \text{DniDirector}, \text{FechaIngresoDirector})}(\text{DEPARTAMENTO})$ .
- ◆ Se puede realizar en dos pasos creando una tabla intermedia:
  - $\text{DEPT} \leftarrow \rho_{S(\text{NombreDpto}, \text{NumDptoProyecto}, \text{DniDirector}, \text{FechaIngresoDirector})}(\text{DEPARTAMENTO})$ .
  - $\text{PROYECTO\_DPTO} \leftarrow \text{PROYECTO} \bowtie \text{DEPT}$ .
- ◆ El atributo *NumDptoProyecto* recibe el nombre de atributo de conexión del *natural join* (notar que solo mantiene uno de los atributos de conexión en la tabla resultante).

PROYECTO\_DPTO

NombreProyecto	<u>NumProyecto</u>	UbicacionProyecto	NumDpto-Proyecto	NombreDpto	DniDirector	FechaIngresoDirector
ProductoX	1	Valencia	5	Investigación	333445555	22-05-1988
ProductoY	2	Sevilla	5	Investigación	333445555	22-05-1988
ProductoZ	3	Madrid	5	Investigación	333445555	22-05-1988
Computación	10	Gijón	4	Administración	987654321	01-01-1995
Reorganización	20	Madrid	1	Sede Central	888665555	19-06-1981
Comunicaciones	30	Gijón	4	Administración	987654321	01-01-1995

# Álgebra Relacional: CONCATENACIÓN EQUIJOIN y NATURAL

- ♦ Si los atributos donde se va a realizar la concatenación natural ya tienen el mismo nombre ya o hace falta hacer el renombramiento de los atributos, como por ejemplo en DEPARTAMENTO y LOCALIZACIONES\_DPTO el atributo NumeroDpto es el mismo en la dos tablas:
  - $\text{LOC\_DPTO} \leftarrow \text{DEPARTAMENTO} \bowtie \text{LOCALIZACIONES\_DPTO}$ .

LOC\_DPTO

NombreDpto	NúmeroDpto	DniDirector	FechaIngresoDirector	Lugar
Sede Central	1	888665555	19-06-1981	Madrid
Administración	4	987654321	01-01-1995	Gijón
Investigación	5	333445555	22-05-1988	Valencia
Investigación	5	333445555	22-05-1988	Sevilla
Investigación	5	333445555	22-05-1988	Madrid

# Álgebra Relacional: CONCATENACIÓN EQUIJOIN y NATURAL

- ♦ La concatenación se puede realizar entre diversas tablas:
  - $((\text{PROYECTO} \bowtie_{\text{NumDptoProyecto=NumeroDpto}} \text{DEPARTAMENTO}) \bowtie_{\text{DniDirector=Dni}} \text{EMPLEADO})$ .

# Álgebra Relacional: Operaciones Adicionales (Proyección Generalizada)

- ◆ Amplia las posibilidades de proyección original: permite funciones de atributos en la lista de proyección:
- ◆  $\pi_{F_1, F_2, \dots, F_n}(R)$ , donde  $F_1, F_2, \dots, F_n$  son funciones sobre los atributos de la relación  $R$ .
- ◆ Supongamos la relación EMPLEADO (Dni, Sueldo, Deducción, Antigüedad) y supongamos que queremos sacar un informe en el cual se muestre:
  - SalarioNeto = Sueldo – Deducción
  - Gratificaciones = 2000 \* Antigüedad
  - Impuestos = 0.25 \* Sueldo
- ◆ Así se puede utilizar una proyección generalizada para sacar este informe:
  - $\text{INFORME} \leftarrow \rho_{(\text{Dni}, \text{SalarioNeto}, \text{Gratificaciones}, \text{Impuestos})} \pi_{\text{Dni}, \text{Sueldo} - \text{Deducción}, 2000 * \text{Antigüedad}, 0.25 * \text{Sueldo}}(\text{EMPLEADO})$ .



# Álgebra Relacional: Operaciones Adicionales (Agregación y Agrupamiento)

- ♦ Se pueden hacer cálculos con los atributos de las relaciones a través de funciones matemáticas de agregación:
  - SUMA (SUM)
  - MEDIA (AVERAGE)
  - MÁXIMO (MAXIMUM)
  - MÍNIMO (MINIMUM)
  - CONTAR (COUNT)
- ♦ Otro tipo de función es la agrupación de atributos de una relación para luego posteriormente aplicar alguna de las funciones de agregación anteriores.
- ♦ La función agregada más general AGRUPA tuplas por el valor de ciertos atributos, y sobre esos grupos puede aplicar operaciones de AGREGACIÓN.
- ♦ Definimos la función agregada (script F) por el símbolo  $\mathfrak{F}$  :
  - $\langle \text{atributos de agrupamiento} \rangle \mathfrak{F} \langle \text{lista de funciones} \rangle (R)$
- ♦ Los  $\langle \text{atributos de agrupamiento} \rangle$  es una lista de atributos de la relación R, y  $\langle \text{lista de funciones} \rangle$  es una lista de parejas ( $\langle \text{función} \rangle, \langle \text{atributo} \rangle$ ).
- ♦ En cada una de las parejas  $\langle \text{función} \rangle$  puede ser cualquiera de las funciones de agregación anteriores, y  $\langle \text{atributo} \rangle$  es un atributo de la relación especificada por R.

# Álgebra Relacional: Operaciones Adicionales (Agregación y Agrupamiento)

- ♦ Por ejemplo recuperar cada número de departamento, el número de empleados del mismo, y la media de sueldos, renombrando los atributos resultantes:

$\rho_{R(Dno, NumEmpleados, MediaSueldos)} (Dno \bowtie COUNT Dni, AVERAGE Sueldo (EMPLEADO)).$

(a)  $\rho_{R(Dno, NumEmpleados, MediaSueldos)} (Dno \bowtie COUNT Dni, AVERAGE Sueldo (EMPLEADO)).$

(b)  $Dno \bowtie COUNT Dni, AVERAGE Sueldo (EMPLEADO).$

(c)  $\bowtie COUNT Dni, AVERAGE Sueldo (EMPLEADO).$

(a) R

Dno	NumEmpleados	MediaSueldos
5	4	33250
4	3	31000
1	1	55000

(b)

Dno	ContarDni	PromedioSueldo
5	4	33250
4	3	31000
1	1	55000

(c)

ContarDni	PromedioSueldo
8	35125

# Álgebra Relacional: Operaciones Adicionales (Agregación y Agrupamiento)

- ♦ Si no se hiciese el renombrado a través de la función  $\rho$ , el nombre de los atributos de la relación de salida se cambia a la forma  $\langle \text{función} \rangle \langle \text{atributo} \rangle$ , podemos verlo en la tabla (b) de la transparencia anterior (esta notación no es estándar, aunque es la notación que utiliza el libro).
- ♦ Si no se especifican atributos de agrupamiento, las funciones de agregación se aplican a todas la tuplas indiscriminadamente de la relación (ver tabla (c) de la transparencia anterior).
- ♦ Es importante recalcar que en general las duplicaciones no se eliminan cuando se aplican funciones de agregación, por lo tanto hay que tener con las interpretaciones de aplicar a estos agrupamientos las funciones de agregación.
- ♦ Si queremos eliminar duplicados antes de pasar una función de agregación aplicamos la función DISTINCT.

# Álgebra Relacional: Operaciones Adicionales (Cierre Recursivo)

- ♦ El cierre recursivo se aplica a una relación recursiva entre tuplas del mismo tipo, como por ejemplo las que se establecen entre un empleado y un supervisor.
- ♦ Esta relación esta descrita por la FK SuperDni de EMPLEADO.
- ♦ Un ejemplo de operación recursiva sería la recuperación de supervisiones de un empleado  $e$  a todos sus niveles, es decir todos los empleados  $e'$  directamente supervisados por  $e$ , los  $e''$  que son todos los supervisados por  $e'$  y así sucesivamente.
- ♦ Como haríamos esto con un ejemplo, para indicar los Dni de todos los empleado  $e'$  supervisados directamente (a nivel 1) por el empleado  $e$  cuyo nombre es Eduardo Ochoa:
- ♦  $\text{DNI\_OCHOA} \leftarrow \pi_{\text{Dni}}(\sigma_{\text{Nombre}='Eduardo' \text{ AND } \text{Apellido1}='Ochoa'} \text{ EMPLEADO}))$
- ♦  $\text{SUPERVISION}(\text{Dni1}, \text{Dni2}) \leftarrow \pi_{\text{Dni}, \text{SuperDni}}(\text{EMPLEADO})$
- ♦  $\text{RESULTADO1}(\text{Dni}) \leftarrow \pi_{\text{Dni1}}(\text{SUPERVISION } |><|_{\text{Dni2}=\text{Dni}} \text{ DNI\_OCHOA})$

# Álgebra Relacional: Operaciones Adicionales (Cierre Recursivo)

- ♦ Si queremos recuperar los supervisados por Ochoa a nivel 2:
- ♦  $\text{RESULTADO2(Dni)} \leftarrow \pi_{\text{Dni1}}(\text{SUPERVISION } |><|_{\text{Dni2=Dni}} \text{ RESULTADO1})$
- ♦ Para obtener los empleados supervisados a nivel 1 y 2 hacemos la UNIÓN:
- ♦  $\text{RESULTADO} \leftarrow \text{RESULTADO2} \cup \text{RESULTADO1}$
- ♦ Podemos seguir así indefinidamente hasta alcanzar todos los niveles , sin emplear un mecanismo de bucle recursivo.
- ♦ En el estándar SQL3 hay una sintaxis exclusiva para el cierre recursivo.

# Álgebra Relacional: Operaciones Adicionales (Cierre Recursivo)

## SUPERVISION

(El DNI de Ochoa es 888665555)

(Dni)

(SuperDni)

Dni1	Dni2
123456789	333445555
333445555	888665555
999887777	987654321
987654321	888665555
666444444	333445555
453453453	333445555
987987987	987654321
888665555	null

## RESULTADO1

Dni
333445555
987654321

(Supervisado por Ochoa)

## RESULTADO2

Dni
123456789
999887777
666884444
453453453
987987987

(Supervisado por subordinados de Ochoa)

## RESULTADO

Dni
123456789
999887777
666884444
453453453
987987987
333445555
987654321

(RESULTADO1  $\cup$  RESULTADO2)

# Álgebra Relacional: Operaciones Adicionales (Concatenación Externa)

- ◆ Las concatenaciones que mantienen solo la tuplas coincidentes (a través de una cierta condición), se llaman concatenaciones internas (son las que hemos visto hasta ahora).
- ◆ Pero podemos querer por ejemplo una lista de todos los empleados junto con los departamentos que controlan (si no controlan que aparezca NULL).
- ◆ Para esta consulta podemos utilizar la **CONCATENACIÓN EXTERNA IZQUIERDA (LEFT OUTER JOIN)**:
  - $TEMP \leftarrow (EMPLEADO \parallel_{Dni=DniDirector} DEPARTAMENTO)$
  - $RESULTADO \leftarrow \pi_{Nombre, Apellido1, Apellido2, NombreDpto}(TEMP)$
- ◆ Ver la siguiente transparencia.
- ◆ Notar que mantiene cada tupla de la primera relación (relación IZQUIERDA), y **si no encuentra ninguna tupla en la segunda relación a través de los atributos de concatenación especificados, coloca un NULL**.
- ◆ La **CONCATENACIÓN EXTERNA DERECHA (RIGHT OUTER JOIN)** es similar,  $\parallel_{>|}$  (mantiene todas la tuplas de la relación derecha).
- ◆ **CONCATENACIÓN EXTERNA COMPLETA (FULL OUTER JOIN)**,  $\parallel_{><||}$ .

# Álgebra Relacional: Operaciones Adicionales (Concatenación Externa)

## RESULTADO

Nombre	Apellido1	Apellido2	NombreDpto
José	Pérez	Pérez	NULL
Alberto	Campos	Sastre	Investigación
Alicia	Jiménez	Celaya	NULL
Juana	Sainz	Oreja	Administración
Fernando	Ojeda	Ordóñez	NULL
Aurora	Oliva	Avezuela	NULL
Luis	Pajares	Morera	NULL
Eduardo	Ochoa	Paredes	Sede Central



# Álgebra Relacional: Operaciones Adicionales (Unión Externa)

- ♦ Fue diseñada para obtener la unión de tuplas no compatibles, cuando hay compatibilidad parcial, por ejemplo
  - $R(X,Y)$  y  $S(X,Z)$ , dando la unión externa otra relación  $T(X,Y,Z)$
  - Cuando alguna parte  $X$  de  $R$  o  $S$  no se encuentran entonces habrá valores NULL en  $Y$  o  $Z$ .
- ♦ Por ejemplo: **ESTUDIANTE(Nombre, Dni, Departamento, Tutor)** y **PROFESOR(Nombre, Dni, Departamento, Cargo)**.
- ♦ Solo comparten los tres primeros atributos, y la unión externa de ambas sería:
  - **ESTUDIANTE\_O\_PROFESOR(Nombre, Dni, Departamento, Tutor, Cargo)**
- ♦ Cuando **ESTUDIANTE** Y **PROFESOR** se combinen en una **UNION EXTERNA**, los tres primeros atributos que provengan de **ESTUDIANTE** tendrán NULL en el Cargo, y cuando provengan de **PROFESOR** tendrán NULL en el atributo Tutor.

# Correspondencia con SQL

- ♦  $\pi_{\text{atributos}} (\sigma_{\text{condición}} (R))$  SELECT *atributos* FROM *R* WHERE *condición*
- ♦  $\rho_{A/C} (\pi_{A,B} (\sigma_{\text{condición}} (R)))$  SELECT A AS C, B FROM *R* WHERE *condición*
- ♦  $S \leftarrow (\pi_{\text{atributos}} (\sigma_{\text{condición}} (R)))$  CREATE VIEW S AS *atributos* FROM *R* WHERE *condición*
- ♦  $\pi_{\text{atributos}} (\sigma_{\text{condición}} (R \bowtie S))$  SELECT *atributos* FROM *R* JOIN *S* WHERE *condición*
- ♦  $\pi_{\text{atributos}} (\sigma_{c_1} (R \bowtie_{c_2} S))$  SELECT *atributos* FROM *R* JOIN *S* WHERE  $c_1$  and  $c_2$
- ♦  $\pi_{\text{atributos}} (\sigma_{\text{condición}} (R \times S))$  SELECT *atributos* FROM *R*, *S* WHERE *condición*
- ♦  $R \cup S, R \cap S, R - S$  *R* UNION *S*, *R* INTERSECT *S*, *R* EXCEPT *S*
- ♦  $\text{atributos } \mathfrak{I} \text{ Count(A), Sum(B)... (R)}$  SELECT Count(A), Sum(B)... FROM *R* GROUP BY *atributos*

# BD Empresa

## EMPLEADO

▲	Nombre text	Apellido1 text	Apellido2 text	Dni integer	FechaNac date	Direccion text	Sexo "char"	Sueldo numeric	SuperDni integer	Dno integer
1	Jose	Perez	Perez	123456789	1965-09-01	Eloy I, 98	H	30000	333445555	5
2	Alberto	Campos	Sastre	333445555	1955-12-08	Avda Rios, 9	H	40000	888665555	5
3	Alicia	Jimenez	Celaya	999887777	1968-05-12	Gran Via, 38	M	25000	987654321	4
4	Juana	Sainz	Oreja	987654321	1941-06-20	Cerquillas, 67	M	43000	888665555	4
5	Eduardo	Ochoa	Paredes	888665555	1937-11-10	Las Peñas, 1	H	55000	[null]	1
6	Fernan...	Ojeda	Ordoñez	666884444	1962-09-15	Portillo, S/N	M	38000	333445555	5
7	Luis	Pajares	Morera	987987987	1969-03-29	Enebros, 90	H	25000	987654321	4
8	Aurora	Oliva	Avezuela	453453453	1972-07-31	Anton, 6	M	25000	333445555	5

## PROYECTO

▲	NombreProyecto text	NumProyecto integer	UbicacionProyecto text	NumDptoProyecto integer
1	PruductoX	1	Valencia	5
2	ProductoY	2	Sevilla	5
3	ProductoZ	3	Madrid	5
4	Computacion	10	Gijon	4
5	Reorganizacion	20	Madrid	1
6	Comunicaciones	30	Gijon	4

## LOCALIZACIONES\_DPTO

▲	NumeroDpto integer	UbicacionDpto text
1	1	Madrid
2	4	Gijon
3	5	Valencia
4	5	Sevilla
5	5	Madrid

## TRABAJA\_EN

▲	DniEmpleado integer	NumProy integer	Horas numeric
1	123456789	1	32.5
2	123456789	2	7.5
3	666884444	3	40.0
4	453453453	1	20.0
5	453453453	2	20.0
6	333445555	2	10.0
7	333445555	10	10.0
8	333445555	3	10.0
9	333445555	20	10.0
10	999887777	30	30.0
11	999887777	10	10.0
12	987987987	10	35.0
13	987987987	30	5.0
14	987654321	30	20.0
15	987654321	20	15.0
16	888665555	20	[null]

## SUBORDINADO

▲	DniEmpleado integer	NombSubordinado text	Sexo "char"	FechaNac date	Relacion text
1	333445555	Alicia	M	1986-04-05	Hija
2	333445555	Teodoro	H	1983-10-25	Hijo
3	333445555	Luisa	M	1958-05-03	Esposa
4	987654321	Alfonso	H	1942-01-28	Esposo
5	123456789	Miguel	H	1988-01-04	Hijo
6	123456789	Alicia	M	1988-12-30	Hija
7	123456789	Elisa	M	1967-05-05	Esposa

## DEPARTAMENTO

▲	NombreDpto text	NumeroDpto integer	DniDirector integer	FechaIngresoDirector date
1	Investigacion	5	333445555	1988-05-22
2	Administracion	4	987654321	1995-01-01
3	Sede Central	1	888665555	1981-06-19

# Campos, Tuplas y Tablas en una BD

## EMPLEADO

Nombre	Apellido1	Apellido2	<u>Dni</u>	FechaNac	Direccion	Sexo	Sueldo	SuperDni	Dno
text	text	text	integer	date	text	"char"	numeric	integer	integer

## DEPARTAMENTO

NombreDpto	<u>NumeroDpto</u>	DniDirector	FechaIngresoDirector
text	integer	integer	date

## LOCALIZACIONES\_DPTO

<u>NumeroDpto</u>	<u>UbicacionDpto</u>
integer	text

## PROYECTO

NombreProyecto	<u>NumProyecto</u>	UbicacionProyecto	NumDptoProyecto
text	integer	text	integer

## TRABAJA\_EN

<u>DniEmpleado</u>	<u>NumProy</u>	Horas
integer	integer	numeric

## SUBORDINADO

<u>DniEmpleado</u>	<u>NombSubordinado</u>	Sexo	FechaNac	Relacion
integer	text	"char"	date	text

# Álgebra Relacional: Ejemplos

**Consulta 1.** Recupere el nombre y la dirección de todos los empleados que trabajan en el departamento 'Investigación'.

$DPTO\_INVESTIGACION \leftarrow \sigma_{\text{NombreDpto}='Investigación'}(DEPARTAMENTO)$

$EMPS\_INVESTIGACION \leftarrow (DPTO\_INVESTIGACION \bowtie_{\text{NúmeroDpto}=\text{Dno}} EMPLEADO)$

$RESULTADO \leftarrow \pi_{\text{Nombre, Apellido1, Dirección}}(EMPS\_INVESTIGACION)$

Como una expresión única, esta consulta se convierte en:

$\pi_{\text{Nombre, Apellido1, Dirección}}(\sigma_{\text{NombreDpto}='Investigación'}(DEPARTAMENTO \bowtie_{\text{NúmeroDpto}=\text{Dno}} EMPLEADO))$

- ♦ Esta consulta se podría expresar de otras formas: se podría invertir el orden de las operaciones de **CONCATENACIÓN** y **SELECCIÓN** y se podría sustituir por una concatenación natural, después por supuesto de hacer un renombrado (ejercicio para casa).

# Álgebra Relacional: Ejemplos

**Consulta 2.** Por cada proyecto ubicado en ‘Gijón’, enumere su número, el número de departamento que lo gestiona y los apellidos, dirección y fecha de nacimiento del director del departamento.

PROYECTOS\_GIJON  $\leftarrow \sigma_{\text{UbicacionProyecto}='Gijón'}(\text{PROYECTO})$

DEPT\_CONTROL  $\leftarrow (\text{PROYECTOS_GIJON} \bowtie_{\text{NumDptoProyecto}=\text{NúmeroDpto}} \text{DEPARTAMENTO})$

DIRECTOR\_DPTO\_PROYECTO?  $(\text{DEPT\_CONTROL} \bowtie_{\text{DniDirector}=\text{Dni}} \text{EMPLEADO})$

RESULTADO  $\leftarrow \pi_{\text{NumProyecto}, \text{NumDptoProyecto}, \text{Apellido1}, \text{Dirección}, \text{FechaNac}}(\text{DIRECTOR\_DPTO\_PROYECTO})$

# Álgebra Relacional: Ejemplos

**Consulta 4.** Haga una lista de los números de proyecto en los que esté involucrado cualquier empleado cuyo primer apellido sea 'Pérez', ya sean trabajadores o directores del departamento que gestiona ese proyecto.

$PEREZ(DniEmpleado) \leftarrow \pi_{Dni}(\sigma_{Apellido1='Pérez'}(EMPLEADO))$   
 $PROYS\_PEREZ \leftarrow \pi_{NumProy}(TRABAJA\_EN \bowtie PEREZ)$   
 $DIRECTORES \leftarrow \pi_{Apellido1, NúmeroDpto}(EMPLEADO \bowtie_{Dni=DniDirector} DEPARTAMENTO)$   
 $DEPTS\_ADMINISTRADOS\_PEREZ(NumDptoProyecto) \leftarrow \pi_{NúmeroDpto}(\sigma_{Apellido1='Pérez'}(DIRECTORES))$   
 $DEPTS\_DIRECTOR\_PEREZ(NumProy) \leftarrow$   
 $\pi_{NumProyecto}(DEPTS\_ADMINISTRADOS\_PEREZ \bowtie PROYECTO)$   
 $RESULTADO \leftarrow (PROYS\_PEREZ \cup DEPTS\_DIRECTOR\_PEREZ)$

Como una única expresión, esta consulta se transforma en

$\pi_{NumProy}(TRABAJA\_EN \bowtie_{DniEmpleado=Dni}(\pi_{Dni}(\sigma_{Apellido1='Pérez'}(EMPLEADO))))$   
 $\cup \pi_{NumProy}((\pi_{NúmeroDpto}(\sigma_{Apellido1='Pérez'}(\pi_{Apellido1, NúmeroDpto}(EMPLEADO))))$   
 $\bowtie_{Dni=DniDirector} DEPARTAMENTO)) \bowtie_{NúmeroDpto=NumDptoProyecto} PROYECTO)$

# Álgebra Relacional: Ejemplos

**Consulta 5.** Liste los nombres de todos los empleados con dos o más subordinados.

Estrictamente hablando, esta consulta no puede llevarse a cabo con el *álgebra relacional básica (original)*. Tenemos que usar la operación FUNCIÓN AGREGADA con la función CONTAR. Asumimos que los asalariados del *mismo* empleado tienen *distinto* valor NOMBRE\_SUBORDINADO.

$$\begin{aligned} T1(\text{Dni}, \text{NumSubordinados}) &\leftarrow \text{DniEmpleado} \bowtie \text{COUNT NombreSubordinado}(\text{SUBORDINADO}) \\ T2 &\leftarrow \sigma_{\text{NumSubordinados} \geq 2}(T1) \\ \text{RESULTADO} &\leftarrow \pi_{\text{Apellido1}, \text{Nombre}}(T2 \bowtie \text{EMPLEADO}) \end{aligned}$$



# Álgebra Relacional: Ejemplos

**Consulta 6.** Recuperar los nombres de los empleados que no tienen subordinados.

Se trata de un ejemplo del tipo de consulta que utiliza la operación MENOS (DIFERENCIA DE CONJUNTOS).

TODOS\_EMPLEADOS  $\leftarrow \pi_{\text{Dni}}(\text{EMPLEADO})$

EMPS\_CON\_SUBORDINADOS(Dni)  $\leftarrow \pi_{\text{DniEmpleado}}(\text{SUBORDINADO})$

EMPS\_SIN\_SUBORDINADOS  $\leftarrow (\text{TODOS\_EMPLEADOS} - \text{EMPS\_CON\_SUBORDINADOS})$

RESULTADO  $\leftarrow \pi_{\text{Apellido1, Nombre}}(\text{EMPS\_SIN\_SUBORDINADOS} \bowtie \text{EMPLEADO})$

Como una única expresión, esta consulta queda del siguiente modo:

$\pi_{\text{Apellido1, Nombre}}((\pi_{\text{Dni}}(\text{EMPLEADO}) - \rho_{\text{Dni}}(\pi_{\text{DniEmpleado}}(\text{SUBORDINADO}))) \bowtie \text{EMPLEADO})$

**Consulta 7.** Enumerar los nombres de los directivos que tienen, al menos, un subordinado.

DIRECTORES(Dni)  $\leftarrow \pi_{\text{DniDirector}}(\text{DEPARTAMENTO})$

EMPS\_CON\_SUBORDINADOS(Dni)  $\leftarrow \pi_{\text{DniEmpleado}}(\text{SUBORDINADO})$

DIRECTORES\_CON\_SUBORDINADOS  $\leftarrow (\text{DIRECTORES} \cap \text{EMPS\_CON\_SUBORDINADOS})$

RESULTADO  $\leftarrow \pi_{\text{Apellido1, Nombre}}(\text{DIRECTORES\_CON\_SUBORDINADOS} \bowtie \text{EMPLEADO})$