

# Álgebra Relacional

El álgebra relacional es muy importante por varias razones. La primera, porque proporciona un fundamento formal para las operaciones del modelo relacional. La segunda razón, y quizá la más importante, es que se utiliza como base para la implementación y optimización de consultas en los RDBMS (Sistemas de administración de bases de datos relacionales, Relational DataBase Management Systems).

Bibliografía:

Fundamentos de Sistemas de Bases de Datos. Ramez Elmasri, Shamkant Navathe, 5ta edición.

BDTema05-mrarcr.pdf

Fundamentos de Bases de Datos. Silberschatz, Korth, Sudarshan, 4ta edición.

# Álgebra Relacional

El álgebra relacional es un lenguaje de consulta procedimental. Consta de un conjunto de operaciones que toman como entrada una o dos relaciones y producen como resultado una nueva relación. Las operaciones fundamentales del álgebra relacional son selección, proyección, unión, diferencia de conjuntos, producto cartesiano y renombramiento. Además de las operaciones fundamentales hay otras operaciones, por ejemplo, intersección de conjuntos, reunión natural, división y asignación. Estas operaciones se definirán en términos de las operaciones fundamentales.

# Algebra Relacional Operaciones

## Grupo 1

Incluye el conjunto de operaciones de la teoría matemática de conjuntos, los cuales son aplicables porque cada relación está definida de modo que sea un conjunto de tuplas en el modelo relacional formal.

UNIÓN (UNION),

INTERSECCIÓN (INTERSECTION),

DIFERENCIA DE CONJUNTOS (SET DIFFERENCE)

PRODUCTO CARTESIANO (CARTESIAN PRODUCT)

# Algebra Relacional Operaciones

## Grupo 2

Operaciones desarrolladas específicamente para las bases de datos relacionales

SELECCIÓN (SELECT)

PROYECCIÓN (PROJECT)

CONCATENACIÓN, COMBINACIÓN (JOIN)

DISTINTO

# Operación unaria SELECCIÓN

La SELECCIÓN se emplea para seleccionar un subconjunto de las tuplas de una relación que satisfacen una condición de selección. Se puede considerar esta operación como un filtro que mantiene sólo las tuplas que satisfacen una determinada condición. La SELECCIÓN puede visualizarse también como una partición horizontal de la relación en dos conjuntos de tuplas, las que satisfacen la condición son seleccionadas y las que no, son descartadas.

$\sigma_{Dno=4}(\text{EMPLEADO})$

$\sigma_{\text{Sueldo}>30000}(\text{EMPLEADO})$

$\sigma_{\langle \text{condición de selección} \rangle}(R)$

# Operación unaria SELECCIÓN

La condición de selección es una expresión lógica especificada sobre los atributos de la relación  $R$ . Observe que  $R$  es, generalmente, una expresión de álgebra relacional cuyo resultado es una relación. El resultado de la SELECCIÓN tiene los mismos atributos que  $R$ .

**Observe que la operación SELECCIÓN es conmutativa**

$$\sigma_{\langle \text{condición1} \rangle}(\sigma_{\langle \text{condición2} \rangle}(R)) = \sigma_{\langle \text{condición2} \rangle}(\sigma_{\langle \text{condición1} \rangle}(R))$$

Por tanto, puede aplicarse una secuencia de SELECCIONES en cualquier orden. Además, siempre podemos combinar una cascada de operaciones SELECCIÓN en una sola a través de un (AND):

$$\begin{aligned} & \sigma_{\langle \text{condición1} \rangle}(\sigma_{\langle \text{condición2} \rangle}(\dots(\sigma_{\langle \text{condición} \rangle}(R))\dots)) \\ &= \sigma_{\langle \text{condición1} \rangle} \text{ AND } \langle \text{condición2} \rangle \text{ AND } \dots \text{ AND } \langle \text{condición} \rangle(R) \end{aligned}$$

# Operación unaria PROYECCIÓN

Si pensamos en una relación como en una tabla, la operación SELECCIÓN elige algunas de las filas de la tabla a la vez que descarta otras. Por otro lado, PROYECCIÓN selecciona ciertas columnas de la tabla y descarta otras. Si sólo estamos interesados en algunos atributos de una relación, usamos la operación PROYECCIÓN para planear la relación sólo sobre esos atributos. Por ejemplo, para listar el nombre, el primer apellido y el sueldo de cada empleado, podemos usar PROYECCIÓN de la siguiente forma:

$$\pi_{\text{Apellido1, Nombre, Sueldo}}(\text{EMPLEADO})$$
$$\pi_{\langle \text{lista de atributos} \rangle}(R)$$

# Operación unaria PROYECCIÓN

El número de tuplas de una relación resultante de una operación PROYECCIÓN es siempre menor o igual que el de las contenidas en R, Si la lista de proyección es una superclave de R (esto es, incluye alguna clave de R) la relación resultante tiene el mismo número de tuplas que R. Además,

$$\pi_{\langle \text{lista1} \rangle} (\pi_{\langle \text{lista2} \rangle} (R)) = \pi_{\langle \text{lista1} \rangle} (R)$$



# Composición de Operaciones

Es importante el hecho de que el resultado de una operación relacional sea también una relación. Considérese la consulta más compleja «Encontrar los clientes que viven en Peguerinos». Hay que escribir:

$$\Pi_{\text{nombre-cliente}} \left( \sigma_{\text{ciudad-cliente} = \text{«Peguerinos»}} (\text{cliente}) \right)$$

# Secuencias de operaciones y la operación unaria ASIGNACIÓN o RENOMBRAR

En general, podemos querer aplicar varias operaciones de álgebra relacional una tras otra. De cualquier forma, podemos escribir las operaciones como una única expresión de álgebra relacional anidando dichas operaciones, o aplicar una sola expresión una única vez y crear relaciones intermedias. En el último caso, puede que queramos asignar nombres a dichas relaciones intermedias, Por ejemplo, para recuperar el nombre, el primer apellido y el sueldo de todos los empleados que trabajan en el departamento 5, debemos aplicar una SELECCIÓN y una PROYECCIÓN. Podemos escribir una expresión de álgebra relacional sencilla de la siguiente forma:

$$\pi_{\text{Nombre, Apellido1, Sueldo}}(\sigma_{\text{Dno}=5}(\text{EMPLEADO}))$$

podemos mostrar la secuencia de operaciones, dando un nombre a cada relación intermedia:

$$\text{DEP5\_EMPS} \leftarrow \sigma_{\text{Dno}=5}(\text{EMPLEADO})$$
$$\text{RESULTADO} \leftarrow \pi_{\text{Nombre, Apellido1, Sueldo}}(\text{DEP5\_EMPS})$$

# Operación unaria ASIGNACIÓN o RENOMBRAR

Podemos definir una operación RENOMBRAR como un operador unario. Una operación RENOMBRAR aplicada a una relación R donde el símbolo  $\rho$  (rho) se utiliza para especificar el operador RENOMBRAR.

# Operación unaria ASIGNACIÓN o RENOMBRAR

Denotado por la letra griega rho minúscula ( $\rho$ ), permite realizar esta tarea. Dada una expresión  $E$  del álgebra relacional, la expresión  $\rho_x(E)$  devuelve el resultado de la expresión  $E$  con el nombre  $x$ . Las relaciones  $r$  por sí mismas se consideran expresiones (triviales) del álgebra relacional. Por tanto, también se puede aplicar la operación renombramiento a una relación  $r$  para obtener la misma relación con un nombre nuevo. Otra forma de la operación renombramiento es la siguiente. Supóngase que una expresión del álgebra relacional  $E$  tiene aridad  $n$ . Por tanto, la expresión  $\rho_x(A_1, A_2, \dots, A_n)(E)$  devuelve el resultado de la expresión  $E$  con el nombre  $x$  y con los atributos con el nombre cambiado a  $A_1, A_2, \dots, A_n$ .

# Operaciones de álgebra relacional de la teoría de conjuntos

## La operación binaria UNIÓN (UNION)

El siguiente grupo de operaciones de álgebra relacional son las correspondientes a la operativa matemática sobre conjuntos. Por ejemplo, para recuperar los DNI de todos los empleados que, o bien trabajan en el departamento 5 o supervisan a éstos, podemos usar la operación UNIÓN del siguiente modo.

$DEP5\_EMPS \leftarrow \sigma_{Dno=5}(EMPLEADO)$

$RESULTADO1 \leftarrow \pi_{Dni}(DEP5\_EMPS)$

$RESULTADO2(Dni). \leftarrow \pi_{SuperDni}(DEP5\_EMPS)$

$RESULTADO \leftarrow RESULTADO1 \cup RESULTADO2$

## La operación binaria **UNIÓN (UNION)**

**UNIÓN.** El resultado de esta operación, especificada como  $R \cup S$ , es una relación que incluye todas las tuplas que están tanto en  $R$  como en  $S$  o en ambas,  $R$  y  $S$ . Las tuplas duplicadas se eliminan.

La operación de **UNION** se realiza entre relaciones compatibles (**UNION COMPATIBLE**)

Para poder realizar la unión,  $R$  y  $S$  deben ser compatibles.

Dos esquemas de relación son compatibles si tienen el mismo grado (misma cantidad de atributos y los dominios de los atributos son equivalentes)

# La operación binaria INTERSECCIÓN (INTERSECT)

INTERSECCIÓN. El resultado de esta operación, especificada como  $R \cap S$ , es una relación que incluye todas las tuplas que están en R y en S.

Observe que tanto la UNIÓN como la INTERSECCIÓN son operaciones conmutativas, esto es,  $R \cup S = S \cup R$  y  $R \cap S = S \cap R$

La UNIÓN y la INTERSECCIÓN pueden tratarse como operaciones de n-array aplicables a cualquier número de relaciones porque son estructuras asociativas, es decir,  $R \cup (S \cup T) = (R \cup S) \cup T$  y  $(R \cap S) \cap T = R \cap (S \cap T)$

Para poder realizar la intersección, R y S deben ser compatibles.

Dos esquemas de relación son compatibles si tienen el mismo grado (misma cantidad de atributos y los dominios de los atributos son equivalentes)

# Operación binaria DIFERENCIA DE CONJUNTO

DIFERENCIA DE CONJUNTO (o MENOS). El resultado de esta operación, especificada como  $R - S$ , es una relación que incluye todas las tuplas que están en  $R$  pero no en  $S$ .

La operación MENOS no es conmutativa; por tanto, en general,

$$R - S \neq S - R$$

Obsérvese que se puede volver a escribir cualquier expresión del álgebra relacional utilizando la intersección de conjuntos sustituyendo la operación intersección por un par de operaciones de diferencia de conjuntos, de la manera siguiente:  $r \cap s = r - (r - s)$  Por tanto, la intersección de conjuntos no es una operación fundamental y no añade potencia al álgebra relacional. Sencillamente, es más conveniente escribir  $r \cap s$  que  $r - (r - s)$ .

Para poder realizar la diferencia,  $R$  y  $S$  deben ser compatibles.

Dos esquemas de relación son compatibles si tienen el mismo grado (misma cantidad de atributos y los dominios de los atributos son equivalentes)



# Operación binaria Producto Cartesiano

Ahora vamos a tratar la operación PRODUCTO CARTESIANO, conocida también como PRODUCTO CRUZADO o CONCATENACIÓN CRUZADA, que se identifica por  $\times$ . Se trata también de una operación de conjuntos binarios, aunque no es necesario que las relaciones en las que se aplica sean una unión compatible. En su forma binaria produce un nuevo elemento combinando cada tupla de una relación conjunto con los de la otra. En general, el resultado de  $R(A_1, A_2, \dots, A_n) \times S(B_1, B_2, \dots, B_m)$  es una relación  $Q$  con un grado de  $n + m$  atributos  $Q(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$ , en este orden. La relación resultante  $Q$  tiene una tupla por cada combinación de éstas (una para  $R$  y otra para  $S$ ). Por tanto, si  $R$  tiene  $|R|$  tuplas (indicado como  $|R| = n_r$ ), y  $S$  cuenta con  $|S|$  tuplas,  $R \times S$  tendrá  $n_r * n_s$  tuplas.

# Operaciones relacionales binarias: CONCATENACIÓN (JOIN)

CONCATENACIÓN, especificada mediante  $\bowtie$ , se emplea para combinar tuplas relacionadas de dos relaciones en una sola. Esta operación es muy importante para cualquier base de datos relacional que cuente con más de una relación, ya que nos permite procesar relaciones entre relaciones. Para ilustrar CONCATENACIÓN, supongamos que queremos recuperar el nombre del director de cada departamento. Para ello, necesitamos combinar las tuplas de departamento y empleado cuyos valores de DniDirector y Dni, respectivamente, sean iguales. Esto se consigue mediante la operación CONCATENACIÓN y extrapolarlo después el resultado sobre los atributos necesarios de la siguiente forma:

$$\text{DIRECTOR\_DPTO} \leftarrow \text{DEPARTAMENTO} \bowtie_{\text{DniDirector}=\text{Dni}} \text{EMPLEADO}$$
$$\text{RESULTADO} \leftarrow \pi_{\text{NombreDpto}, \text{Apellido1}, \text{Nombre}}(\text{DIRECTOR\_DPTO})$$

# Operaciones relacionales binarias: CONCATENACIÓN (JOIN)

La CONCATENACIÓN puede ser enunciada en términos de un PRODUCTO CARTESIANO seguido de una SELECCIÓN. Sin embargo, CONCATENACIÓN es muy importante porque se utiliza con mucha frecuencia cuando se definen consultas a la base de datos. Considere el ejemplo que mostramos anteriormente para explicar el PRODUCTO CARTESIANO, el cual incluía la siguiente secuencia de operaciones:

```
EMPLEADOS_SUBORDINADOS ← NOMBRES_EMPLEADOS × SUBORDINADO
SUBORDINADOS_ACTUALES ←  $\sigma_{\text{Dni}=\text{DniEmpleado}}$ (EMPLEADOS_SUBORDINADOS)
```

Esas dos operaciones pueden sustituirse por una única operación CONCATENACIÓN de la siguiente forma:

```
SUBORDINADOS_ACTUALES ← NOMBRES_EMPLEADOS  $\bowtie_{\text{Dni}=\text{DniEmpleado}}$  SUBORDINADO
```

# Operaciones relacionales binarias: CONCATENACIÓN (JOIN)

La forma general de una CONCATENACIÓN en dos relaciones  $R(A_1, A_2, \dots, A_n)$  Y  $S(B_1, B_2, \dots, B_m)$  es:

$$R \bowtie_{\langle \text{condición de conexión} \rangle} S$$

# Operaciones relacionales binarias: CONCATENACIÓN (JOIN)

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; Q tiene una tupla por cada combinación de éstas (una para R y otra para S) siempre que dicha combinación satisfaga la condición de conexión. Ésta es la principal diferencia existente entre el PRODUCTO CARTESIANO Y la CONCATENACIÓN. En la CONCATENACIÓN sólo aparecen en el resultado las combinaciones de tuplas que satisfacen la condición de conexión, mientras que en el PRODUCTO CARTESIANO se incluyen todas las combinaciones de tuplas. La condición de conexión está especificada sobre los atributos de las dos relaciones R y S y es evaluada para cada combinación de tuplas, incluyéndose en la relación Q resultante en forma de una única tupla combinada sólo aquellas cuya condición de conexión se evalúe como VERDADERO. Una condición general de conexión tiene la forma:

-  
<condición> **AND** <condición> **AND** . . . **AND** <condición>

# Variaciones de CONCATENACIÓN: EQUIJOIN y CONCATENACIÓN NATURAL (NATURAL JOIN)

El uso más habitual de CONCATENACIÓN supone el uso de condiciones de conexión sólo con comparaciones de igualdad, en cuyo caso recibe el nombre de EQUIJOIN. Observe que en el resultado de una EQUIJOIN siempre tenemos uno o más pares de atributos que cuentan con valores idénticos en cada tupla. Por ejemplo, DniDirector y Dni tienen valores idénticos en cada tupla de DIRECTOR\_DPTO debido a la condición de conexión especificada ente estos atributos. Ya que uno de estos valores idénticos es innecesario, se creó una nueva operación llamada CONCATENACIÓN NATURAL (identificada por \*) para deshacerse del segundo atributo superfluo en una condición EQUIJOIN. La definición estándar de esta operación precisa que los dos atributos de conexión tengan el mismo nombre en ambas relaciones. Si éste no es el caso, se aplica en primer lugar una operación de renombrado. En el siguiente ejemplo, primero renombramos el atributo NúmeroDpto de DEPARTAMENTO como NumDptoProyecto, con lo que éste y el de PROYECTO tendrán el mismo nombre, y después se aplica CONCATENACIÓN NATURAL:

# Variaciones de CONCATENACIÓN: EQUIJOIN y CONCATENACIÓN NATURAL (NATURAL JOIN)

$PROYECTO\_DPTO \leftarrow .PROYECTO \rho_{(NombreDpto, NumDptoProyecto, DniDirector, FechaIngresoDirector)}(DEPARTAMENTO)$

Se puede realizar la misma consulta en dos pasos creando una tabla intermedia DEPT de la siguiente forma:

$DEPT \leftarrow \rho_{(NombreDpto, NumDptoProyecto, DniDirector, FechaIngresoDirector)}(DEPARTAMENTO)$

$PROYECTO\_DPTO \leftarrow PROYECTO * DEPT$

El atributo NumDptoProyecto recibe el nombre de atributo de conexión. En la relación PROYECTO\_DPTO, cada tupla combina una de tipo PROYECTO con otra de DEPARTAMENTO para el departamento que controla el proyecto, aunque sólo se mantiene un atributo de conexión. Si los atributos en los que se aplicará la concatenación natural ya tienen el mismo nombre en ambas relaciones, el renombrado es innecesario. Por ejemplo, para aplicar esta operación en los atributos NúmeroDpto de DEPARTAMENTO y LOCALIZACIONES\_DPTO, es suficiente con escribir:

$LOC\_DPTO \leftarrow . DEPARTAMENTO * LOCALIZACIONES\_DPTO$

# Variaciones de CONCATENACIÓN: EQUIJOIN y CONCATENACIÓN NATURAL (NATURAL JOIN)

De forma general, aunque no estandarizada, ésta es una definición de CONCATENACIÓN NATURAL:

$$Q \leftarrow R *_{(\langle \text{lista1} \rangle), (\langle \text{lista2} \rangle)} S$$

En este caso,  $\langle \text{lista1} \rangle$  especifica una lista de  $i$  atributos de  $R$ , mientras que  $\langle \text{lista2} \rangle$  especifica una lista de  $i$  atributos de  $S$ . Estas listas se emplean para formar condiciones de comparación coherentes entre los atributos correspondientes para, a continuación, evaluarlas juntas mediante un operador AND. Sólo se mantiene en el resultado  $Q$  la lista de atributos correspondiente a la primera relación  $R$   $\langle \text{lista1} \rangle$ .



# Variaciones de CONCATENACIÓN: EQUIJOIN y CONCATENACIÓN NATURAL (NATURAL JOIN)

Observe que si ninguna combinación de tuplas satisface la condición de conexión, el resultado de una CONCATENACIÓN es una relación vacía. En general, si R tiene  $n_r$  tuplas y S  $n_s$ , el resultado de una operación de CONCATENACIÓN R I><I condición de conexión S tendrá entre cero Y  $n_r * n_s$  tuplas. El tamaño estimado del resultado dividido entre el valor  $n_r * n_s$  máximo da como resultado un cociente llamado selectividad de concatenación (join selectivity), que es una propiedad de cada condición de conexión. Si no existe ninguna de ellas, todas las combinaciones de tuplas cualificadas y la CONCATENACIÓN degenera en un PRODUCTO CARTESIANO, llamado también PRODUCTO CRUZADO o CONCATENACIÓN CRUZADA.

# Variaciones de CONCATENACIÓN: EQUIJOIN y CONCATENACIÓN NATURAL (NATURAL JOIN)

Como podemos ver, la CONCATENACIÓN se emplea para combinar datos procedentes de múltiples relaciones, de forma que la información pueda presentarse en una única tabla. Estas operaciones se conocen también como concatenaciones internas (inner join) para distinguirlas de una variación llamada concatenaciones externas (outer join). Informalmente, una concatenación interna es un tipo de operación de correspondencia y asociación definida formalmente como una combinación de un PRODUCTO CARTESIANO Y una SELECCIÓN. Una concatenación externa es otra versión más permisiva de la otra. Observe que puede especificarse una concatenación entre una relación y ella misma. La CONCATENACIÓN NATURAL o la EQUIJOIN pueden establecerse también entre múltiples tablas, lo que lleva a una concatenación de n-vías. Por ejemplo, considere la siguiente concatenación de tres vías:

$((\text{PROYECTO} \bowtie_{\text{NúmeroDptoProyecto}=\text{NúmeroDpto}} \text{DEPARTAMENTO}) \bowtie_{\text{DniDirector}=\text{Dni}} \text{EMPLEADO})$

Esta operación enlaza cada proyecto con el departamento al que pertenece para, a continuación, relacionarlo con su director. La malla resultante es una relación consolidada en la que cada tupla contiene esta información proyecto-departamento-director.

# Variaciones de CONCATENACIÓN: EQUIJOIN y CONCATENACIÓN NATURAL (NATURAL JOIN)

## outer join

Mantiene todas las tuplas de R, o todas las tuplas de S, o todas las tuplas de R y S independientemente si mapean las tuplas de R con la de S en sus atributos comunes, cuando no mapean se completan los atributos en la relación resultante con valores nulos.

Los JOIN externos son de tres tipos, Left outer join, Right outer join, Full outer join.

## Inner join

Mantiene las tuplas de R y S que mapean en los atributos comunes a R y S.

Ejemplo: Nombres de actores y de sus agencias representantes, si tienen

$AGEN(codAg, nomAg, direccion, telefono) \leftarrow AGENCIA$

$TEMP \leftarrow (ACTOR \bowtie_{agencia=codAg} AGEN)$

$RESULTADO \leftarrow \prod_{nombre, nomAg} (TEMP)$

# Reunión Externa (outer join)

## OTRAS OPERACIONES DEL A.R. REUNIÓN EXTERNA

### REUNIÓN EXTERNA IZQUIERDA $R = A \bowtie B$

- Conserva en R todas las tuplas de A
- Si no encuentra una tupla coincidente en B, cada atributo de R (correspondiente a B) es NULO

### REUNIÓN EXTERNA DERECHA $R = A \ltimes B$

- Conserva en R todas las tuplas de B
- Si no encuentra una tupla coincidente en A, cada atributo de R (correspondiente a A) es NULO

Ejemplo: Nombres de agencias y de los actores a los que representan;  
incluyendo las agencias que no representan a ningún actor

### REUNIÓN EXTERNA COMPLETA $R = A \ltimes \bowtie B$

- Conserva en R todas las tuplas de A y B
- cuando no encuentra tuplas coincidentes, rellenando con NULO

# Left outer join

The image displays three screenshots of the SQL Server Enterprise Manager interface, illustrating the execution of a left outer join query. Each window shows the SQL statement and the resulting data grid.

**Window 1 (sel R):** Shows the query `1 SELECT R1, R2` and `2 FROM R`. The result grid contains 6 rows of data from table R:

	R1	R2
1	a	a
2	a	b
3	a	c
4	b	a
5	b	b
6	b	c

**Window 2 (sel R):** Shows the query `1 SELECT r.*,s.*` and `2 FROM R left outer join S on R.R1 = s.s1`. The result grid contains 9 rows, including rows with null values for columns S1 and S2:

	R1	R2	S1	S2
1	a	a	a	a
2	a	a	a	b
3	a	b	a	a
4	a	b	a	b
5	a	c	a	a
6	a	c	a	b
7	b	a	[null]	[null]
8	b	b	[null]	[null]
9	b	c	[null]	[null]

**Window 3 (sel S):** Shows the query `1 SELECT S1, S2` and `2 FROM S`. The result grid contains 2 rows of data from table S:

	S1	S2
1	a	a
2	a	b

The status bars at the bottom of each window indicate the number of rows fetched and transaction status: 6 row(s) fetched, 9 row(s) fetched, and 2 row(s) fetched.

# Right outer join

SQL editor window 'sel R' showing a query and its result set.

```
1 SELECT R1, R2
2 FROM R
```

	R1	R2
1	a	a
2	a	b
3	a	c
4	b	a
5	b	b
6	b	c

sysdba@localh... 6 row(s) fetched 1:12 Transa...

SQL editor window 'sel R' showing a right outer join query and its result set.

```
1 SELECT r.*,s.*
2 FROM R right outer join S on R.R1 = s.s1
```

	R1	R2	S1	S2
1	a	a	a	a
2	a	b	a	a
3	a	c	a	a
4	a	a	a	b
5	a	b	a	b
6	a	c	a	b

sysdba@localhost:C:\grchere\bjprj\spl... 6 row(s) fetched 2:13 Transaction started

SQL editor window 'sel S' showing a query and its result set.

```
1 SELECT S1, S2
2 FROM S
```

	S1	S2
1	a	a
2	a	b

sysdba@lo... 2 row(s) fetched 2:7 Trans...

# Full outer join

The image displays three screenshots of a SQL client interface, showing the execution of queries and their results.

**Left Screenshot:** Shows a query window with the following SQL code:

```
1 SELECT R1, R2
2 FROM R
```

The results pane shows a table with 6 rows:

	R1	R2
1	a	a
2	a	b
3	a	c
4	b	a
5	b	b
6	b	c

The status bar indicates "6 row(s) fetched".

**Middle Screenshot:** Shows a query window with the following SQL code:

```
1 SELECT r.*,s.*
2 FROM R full outer join S on R.R1 = s.s1
```

The results pane shows a table with 9 rows:

	R1	R2	S1	S2
1	a	a	a	a
2	a	b	a	a
3	a	c	a	a
4	a	a	a	b
5	a	b	a	b
6	a	c	a	b
7	b	a	[null]	[null]
8	b	b	[null]	[null]
9	b	c	[null]	[null]

The status bar indicates "9 row(s) fetched".

**Right Screenshot:** Shows a query window with the following SQL code:

```
1 SELECT S1, S2
2 FROM S
```

The results pane shows a table with 2 rows:

	S1	S2
1	a	a
2	a	b

The status bar indicates "2 row(s) fetched".

# La operación DIVISIÓN

La DIVISIÓN, especificada mediante  $\div$ , es útil para cierto tipo de consultas que a veces se realizan en aplicaciones de bases de datos, Un ejemplo es, Recuperar los nombres de los empleados que trabajan en todos los proyectos en los que también lo haga 'José Pérez', Para expresar esta consulta usando una DIVISIÓN, proceda del siguiente modo, Primero, recupere en la relación intermedia PEREZ\_PNOS la lista de números de proyecto en los que trabaja 'José Pérez':

$$\text{PEREZ} \leftarrow \sigma_{\text{Nombre}='Jo'se' \text{ AND } \text{Apellido1}='Pérez'}(\text{EMPLEADO})$$
$$\text{PEREZ\_PNOS} \leftarrow \pi_{\text{NumProy}}(\text{TRABAJA\_EN} \bowtie_{\text{DniEmpleado}=\text{Dni}} \text{PEREZ})$$

A continuación, cree una relación que incluya una tupla siempre que el empleado cuyo Dni es DniEmpleado trabaje en el proyecto cuyo número es NumProy en la relación intermedia DNI PNOS:

$$\text{DNI\_PNOS} \leftarrow \pi_{\text{DniEmpleado}, \text{NumProy}}(\text{TRABAJA\_EN})$$

Por último, aplique la DIVISIÓN a ambas relaciones, lo que nos facilita los DNI de los empleados que queremos:

$$\text{DNIS}(\text{Dni}) \leftarrow \text{DNI\_PNOS} \div \text{PEREZ\_PNOS}$$
$$\text{RESULTADO} \leftarrow \pi_{\text{Nombre}, \text{Apellido1}}(\text{DNIS} * \text{EMPLEADO})$$



# La operación DIVISIÓN

La operación DIVISIÓN se aplica a dos relaciones  $R(Z) / S(X)$ , donde  $X \subseteq Z$ .

Permite  $Y = Z - X$  (y, por tanto,  $Z = X \cup Y$ ); es decir, consiente que  $Y$  sea el conjunto de atributos de  $R$  que no lo son de  $S$ . El resultado de una DIVISIÓN es una relación  $T(Y)$  que incluye una tupla  $t$  si las tuplas  $t$   $R$  aparecen en  $R$  con  $t$   $R$   $[Y] = t$ , y con  $t$   $R$   $[X] = t$   $S$  para cada tupla  $t$   $S$  en  $S$ . Esto significa que, para que una tupla  $t$  aparezca en el resultado  $T$  de la DIVISIÓN, los valores de aquélla deben aparecer en  $R$  en combinación con cada tupla en  $S$ . Observe que en la formulación de la operación DIVISIÓN, las tuplas de la relación denominador restringen la relación numerador seleccionando aquellas tuplas del resultado que sean iguales a todos los valores presentes en el denominador. No es necesario saber qué valores son los que están presentes.

La DIVISIÓN puede expresarse como una secuencia de operaciones  $\pi$ ,  $\times$  y  $-$  del siguiente modo:

$$T1 \leftarrow \pi_Y(R)$$

$$T2 \leftarrow \pi_Y((S \times T1) - R)$$

$$T \leftarrow T1 - T2$$

**Figura 6.8.** La operación DIVISIÓN. (a) Dividiendo DNI\_PNOS entre PEREZ\_PNOS. (b)  $T \leftarrow R \div S$ .

(a)

**DNI\_PNOS**

DniEmpleado	NumProy
123456789	1
123456789	2
666884444	3
453453453	1
453453453	2
333445555	2
333445555	3
333445555	10
333445555	20
999887777	30
999887777	10
987987987	10
987987987	30
987654321	30
987654321	20
888665555	20

**PEREZ\_PNOS**

NumProy
1
2

**DNIS**

Dni
123456789
453453453

(b)

**R**

A	B
a1	b1
a2	b1
a3	b1
a4	b1
a1	b2
a3	b2T
a2	b3
a3	b3
a4	b3
a1	b4

**S**

A
a1
a2
a3

**T**

B
b1

Muchas Gracias, hasta  
aquí llegamos hoy.