Álgebra Relacional

José Ramón Paramá Gabía

Capítulo 4

Algebra relacional

Ya hemos visto la estructura y las restricciones del modelo relacional, ahora pasamos a abordar la parte del modelo relacional que nos falta, el conjunto de operaciones para manipular los datos. Codd estableció dos lenguajes de consulta: el *álgebra relacional* y el *cálculo relacional*.

El álgebra relacional es un lenguaje hasta cierto punto procedimental, mientras que el cálculo relacional es un lenguaje no procedimental.

En este capítulo nos centramos en el álgebra relacional, que está formado por un conjunto de operaciones que permiten al usuario especificar peticiones de recuperación. El resultado de una recuperación es una nueva relación, que se ha formado a partir de una o más relaciones. Por lo tanto, las operaciones del álgebra relacional producen nuevas relaciones que podrán manipularse más adelante, utilizando operaciones de la misma álgebra. Una secuencia de operaciones del álgebra relacional forma una *expresión del álgebra relacional*, cuyo resultado será también una relación.

Las operaciones del álgebra relacional suelen clasificarse en dos grupos. Uno contiene el conjunto de operaciones de la teoría de conjuntos (es posible aplicarlas porque las relaciones se definen como conjuntos de tuplas). Entre las operaciones de conjuntos están la UNIÓN, la INTERSECCIÓN, la DIFERENCIA y el PRODUCTO CARTESIANO. El otro grupo consiste en operaciones creadas específicamente para bases de datos relacionales, incluye la SELECCIÓN, la PROYECCIÓN y la REUNIÓN (JOIN) entre otras.

4.1. Selección

La operación de SELECCIÓN sirve para seleccionar un *subconjunto* de las tuplas de una relación que satisfacen una *condición de selección*. Se puede considerar la operación de SELECCIÓN como un filtro que mantiene únicamente aquellas tuplas que satisfacen una condición de cualificación. Por ejemplo, para seleccionar de la relación *empleado* los empleados que trabajan en el departamento 2 o aquellas cuyo salario mínimo rebasa los 20.000 €. Podemos especificar individualmente cada una de estas dos condiciones con la operación SELECCIÓN como sigue:

$$\sigma_{Num-Dept=4}(EMPLEADO)$$
$$\sigma_{Salario>20000}(EMPLEADO)$$

En general, denotamos la operación SELECCIÓN con:

$$\sigma_{<codicion\ de\ seleccion>}(R)$$

donde el símbolo σ (sigma) denota al operador SELECCIÓN, y la condición de selección es una expresión booleana especificada en términos de los atributos de la relación R. Nótese que R normalmente es una *expresión de álgebra relacional* cuyo resultado es una relación. La relación que resulta de la operación SELECCIÓN tiene los mismos atributos que R. La expresión booleana especificada en la condición de selección se compone de una o más *cláusulas* de la forma:

<nombre de atributo> <operador de comparación> <valor constante>, o
<nombre de atributo> <operador de comparación> <nombre de atributo>

donde <nombre de atributo> es el nombre de un atributo de R, <operador de comparación> es normalmente uno de los operadores $\{=, <, \leq, >, \geq, \neq\}$, y <valor constante> es un valor constante del dominio del atributo. Las cláusulas pueden conectarse arbitrariamente mediante los operadores booleanos AND, OR y NOT para formar condiciones compuestas. Por ejemplo, si queremos seleccionar los empleados que trabajan en el departamento 2 y ganan más de 35.000 € al año, o que trabajan en el departamento 1 y ganan más de 25.000 €, podemos escribir la consulta en álgebra relacional como sigue:

$$\sigma_{(Num-Dept=2\ AND\ Salario>35000)\ OR\ (Num-Dept=1\ AND\ Salario>25000)}(EMPLEADO)$$

El resultado se puede observar en la Figura 4.1(a).

(a)

Nombre	Apellido1	Apellido2	NSS	F-Nac	Dirección	Sexo	Salario	Supervisor	Num-Dept
Ana	Moreira	González	5874	25-7-85	Percebe 1	Mujer	38000	5544	2
María	Pérez	Mosquera	5544	21-8-69	Lechuga 6	Mujer	50000	Nulo	2
Luis	Izquierdo	Sánchez	1122	9-9-70	Calcetín 5	Varón	30000	Nulo	1

(b)

Nombre	Apellido1	Apellido2	Salario
José	López	Gómez	34000
Ana	Moreira	González	38000
María	Pérez	Mosquera	50000
Pedro	González	Ruíz	25000
Luis	Izquierdo	Sánchez	30000

(c)

Sexo	Num-Dept
Varón	2
Mujer	2
Varón	1

Figura 4.1: Resultados de las operaciones SELECCIÓN y PROYECCIÓN.

En general, el resultado de una operación SELECCIÓN se determina como sigue. Se aplica la <condición de selección> *independientemente a cada tupla* t de la relación R. Esto se hace sustituyendo cada ocurrencia de un atributo A_i en la condición de selección por su valor en la tupla $t[A_i]$. Si el resultado de evaluar la condición es "verdadero", se *seleccionará* la tupla t.

El operador SELECCIÓN es unario, es decir, se aplica a una sola relación. Además, la operación de selección se aplica a *cada tupla* individualmente, por ello las condiciones de selección no pueden implicar a más de una tupla. El *grado* de la relación resultante de una operación SELECCIÓN es el mismo que el de R.

La operación SELECCIÓN es *conmutativa*, es decir:

$$\sigma_{<cond1>}(\sigma_{<cond2>}(R)) = \sigma_{<cond2>}(\sigma_{<cond1>}(R))$$

Así pues, podemos aplicar una secuencia de operaciones SELECCIÓN en cualquier orden.

4.2. Proyección

Si pensamos en una relación como una tabla, la operación SELECCIÓN selecciona algunas *filas* de la tabla y desecha otras. La operación PROYECCIÓN, en cambio, selecciona ciertas *columnas* de la tabla y desecha las demás. Si sólo nos interesan ciertos atributos de una relación, usamos la operación PROYECCIÓN para "proyectar" la relación sobre esos atributos únicamente. Por ejemplo, si queremos hacer una lista con el nombre, apellidos y el salario de todos los empleados, podemos usar la siguiente operación PROYECCIÓN:

$$\pi_{Nombre,\ Apellido1,\ Apellido2,\ Salario}(EMPLEADO)$$

La relación resultante se muestra en la Figura 4.1(b). La forma general de la operación PROYECCIÓN es:

$$\pi_{<lista\ de\ atributos>}(R)$$

donde π es el símbolo usado para representar la operación PROYECCIÓN y <lista de atributos> es una lista de atributos de la relación R. De nuevo nótese que R es, en general, una *expresión de álgebra relacional* cuyo resultado es una relación (siempre es así), la cual en el caso más simple es únicamente el nombre de una relación. El resultado de una operación PROYECCIÓN contiene únicamente los atributos especificados en la <lista de atributos> y en el mismo orden que aparecen en la lista. Por ello, su grado es igual al número de atributos en la <lista de atributos>.

Si la lista de atributos sólo contiene atributos no clave de R, es probable que aparezcan tuplas repetidas en el resultado. La operación PROYECCIÓN elimina cualquier tupla repetida, así que el resultado de la operación PROYECCIÓN es un *conjunto* de tuplas y por tanto una relación válida. Por ejemplo, consideremos la siguiente operación PROYECCIÓN:

$$\pi_{Sexo,\ Num-Dept}(EMPLEADO)$$

El resultado se muestra en la Figura 4.1(c). Nótese que las tuplas <Mujer, 2> y <Varón, 1> sólo aparecen una vez en dicha figura, aunque su combinación de valores aparece en ambos casos varias veces en la relación EMPLEADO.

La operación PROYECCIÓN no es conmutativa y además:

$$\pi_{<lista1>}(\pi_{lista2}(R)) = \pi_{<lista1>}(R)$$

siempre que <lista2> contenga los atributos que están en <lista1>.

4.3. Secuencia de operaciones y la operación Renombrar

Las relaciones que aparecen en la Figura 4.1 carecen de nombres. En general, es posible que deseemos aplicar varias operaciones de álgebra relacional una tras otra. Para ello, podemos escribir las operaciones en una sola *expresión de álgebra relacional* anidándolas, o bien podemos aplicar una operación cada vez y crear relaciones que contienen los resultados intermedios. En el segundo caso, tendremos que dar nombre a las relaciones que contienen los resultados intermedios. Por ejemplo, si queremos obtener el nombre, los apellidos y el salario de todos los empleados que trabajan en el departamento número 2, debemos aplicar una operación SELECCIÓN y una operación PROYECCIÓN. Podemos escribir una sola expresión de álgebra relacional, de la siguiente forma:

$$\pi_{Nombre,\ Apellido1,\ Apellido2,\ Salario}(\sigma_{Num-Dept=2}(EMPLEADO))$$

La Figura 4.2 muestra el resultado que produce esta expresión. Como alternativa, podemos mostrar explícitamente la secuencia de operaciones, dando un nombre a cada una de las relaciones intermedias:

$$EMPS-DEPT \leftarrow \sigma_{Num-Dept=2}(EMPLEADO)$$
$$RESULTADO \leftarrow \pi_{Nombre,\ Apellido1,\ Apellido2,\ Salario}(EMPS-DEPT)$$

A menudo es más sencillo descomponer una secuencia compleja de operaciones especificando relaciones intermedias que escribir una única expresión. Las dos operaciones anteriores se ilustran en la Figura 4.2(b).

(a)

Nombre	Apellido1	Apellido2	Salario
José	López	Gómez	34000
Ana	Moreira	González	38000
María	Pérez	Mosquera	50000
Pedro	González	Ruíz	25000
Luis	Izquierdo	Sánchez	30000

(b)

EMPS-DEPT									
Nombre	Apellido1	Apellido2	NSS	F-Nac	Dirección	Sexo	Salario	Supervisor	Num-Dept
José	López	Gómez	1245	21-1-71	Real 8	Varón	34000	5544	2
Ana	Moreira	González	5874	25-7-85	Percebe 1	Mujer	38000	5544	2
María	Pérez	Mosquera	5544	21-8-69	Lechuga 6	Mujer	50000	Nulo	2

RESULTADO			
Nombre	Apellido1	Apellido2	Salario
José	López	Gómez	34000
Ana	Moreira	González	38000
María	Pérez	Mosquera	50000

Figura 4.2: Utilización de la operación RENOMBRAR.

También se pueden renombrar los nombres de los atributos, esto puede ser útil en ciertos casos. Para renombrar los atributos de una relación, bastará con que incluyamos una lista con los nuevos nombres de atributos entre paréntesis, como en el siguiente ejemplo:

$$TEMP \leftarrow \sigma_{Num-Dept=2}(EMPLEADO)$$
$$R(Nomb-Pila,\ Primer-Apell,\ Segundo-Apell,\ Sal) \leftarrow \pi_{Nombre,\ Apellido1,\ Apellido2,\ Salario}(TEMP)$$

4.4. Operaciones de la teoría de conjuntos

El siguiente grupo de operaciones del álgebra relacional son las operaciones matemáticas estándar de conjuntos. Por ejemplo, para obtener los números de la seguridad social de todos los empleados que trabajan en el departamento 2 o que supervisan directamente a un empleado que trabaja en dicho departamento, podemos utilizar la operación UNIÓN como sigue:

$$EMPS-DEP2 \leftarrow \sigma_{Num-Dept=2}(EMPLEADO)$$
$$RESULTADO1 \leftarrow \pi_{NSS}(EMPS-DEP2)$$
$$RESULTADO2(NSS) \leftarrow \pi_{Supervisor}(EMPS-DEP2)$$
$$RESULTADO \leftarrow RESULTADO1 \cup RESULTADO2$$

La relación RESULTADO1 contiene los números de la seguridad social de todos los empleados que trabajan en el departamento 2, y RESULTADO2 contiene los números de la seguridad social de todos los empleados que supervisan directamente a empleados que trabajan en el departamento 2. La operación de UNIÓN produce las tuplas que están en RESULTADO1, en RESULTADO2 o en ambas (véase Figura 4.3).

Empleado

Nombre	Apellido1	Apellido2	NSS	F-Nac	Dirección	Sexo	Salario	Supervisor	Num-Dept
José	López	Gómez	1245	21-1-71	Real 8	Varón	34000	5544	2
Ana	Moreira	González	5874	25-7-85	Percebe 1	Mujer	38000	5544	2
María	Pérez	Mosquera	5544	21-8-69	Lechuga 6	Mujer	50000	8877	2
Pedro	González	Ruíz	8811	7-7-79	Puerro 10	Varón	25000	1122	1
Luis	Izquierdo	Sánchez	1122	9-9-70	Calcetín 5	Varón	30000	Nulo	1
Raquel	Núñez	Izquierdo	8877	8-8-80	Rosa, 8	Mujer	30000	1122	1

RESULTADO1

NSS
1245
5874
5544

RESULTADO2

NSS
5544
5544
8877

RESULTADO

NSS
1245
5874
5544
8877

Figura 4.3: Ejemplo del uso de la operación UNIÓN.

De las operaciones de conjuntos clásicas, en el modelo relacional se utilizan varias: UNIÓN, INTERSECCIÓN, DIFERENCIA Y PRODUCTO CARTESIANO. Todas estas operaciones son binarias, es decir, se aplican a dos conjuntos. Al adaptar estas operaciones al modelo relacional, en el caso de la UNIÓN, INTERSECCIÓN y la DIFERENCIA, las dos relaciones de entrada deben ser del mismo *tipo de tuplas*, esta condición se denomina *compatibilidad con la unión* o *unión compatibles*. Se dice que dos relaciones $R(A_1, A_2, \ldots, A_n)$ y $S(B_1, B_2, \ldots, B_n)$ son *unión compatibles* si ambas tienen el mismo grado n y si $\mathrm{dom}(A_i)=\mathrm{dom}(B_i)$ para $1 \leq i \leq n$. Esto significa que las dos relaciones tienen el mismo número de atributos y que cada par de atributos correspondientes tienen el mismo dominio.

Podemos definir las tres operaciones UNIÓN, INTERSECCIÓN y DIFERENCIA para dos relaciones unión compatibles R y S como sigue:

- Unión: el resultado de la operación, denotando por $R \cup S$, es una relación que incluye todas las tuplas que están en R o en S o en ambas. Las tuplas repetidas se eliminan.
- Intersección: el resultado de esta operación, denotado por $R \cap S$, es una relación que incluye as tuplas que están tanto en R como en S.
- Diferencia: el resultado de esta operación, denotado por $R - S$, es una relación que incluye todas las tuplas de R que no están en S.

Adoptaremos la convención de que la relación resultante tiene los mismos nombres de atributos que la primera relación R. La Figura 4.4 ilustra las tres operaciones. Las relaciones ALUMNO y PROFESOR (*instructor* en la figura) de la Figura 4.4(a) son compatibles con la unión, y sus tuplas representan los nombres y primer apellido de alumnos y profesores respectivamente. El resultado de la Unión se muestra en la Figura 4.4(b), el de la Intersección en la Figura 4.4(c) y, la Diferencia ALUMNO-PROFESOR en 4.4(d) y PROFESOR-ALUMNO en 4.4(e).

(a)

STUDENT	FN	LN
	Susan	Yao
	Ramesh	Shah
	Johnny	Kohler
	Barbara	Jones
	Amy	Ford
	Jimmy	Wang
	Ernest	Gilbert

INSTRUCTOR	FNAME	LNAME
	John	Smith
	Ricardo	Browne
	Susan	Yao
	Francis	Johnson
	Ramesh	Shah

(b)

FN	LN
Susan	Yao
Ramesh	Shah
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert
John	Smith
Ricardo	Browne
Francis	Johnson

(c)

FN	LN
Susan	Yao
Ramesh	Shah

(d)

FN	LN
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert

(e)

FNAME	LNAME
John	Smith
Ricardo	Browne
Francis	Johnson

Figura 4.4: Ilustración de las operaciones Unión, Intersección y Diferencia.

Como se puede observar las operaciones Unión e Intersección son conmutativas, mientras que la Diferencia no lo es.

La otra operación de conjuntos que incorpora el álgebra relacional es el Producto Cartesiano. También es una operación binaria de conjuntos, pero las relaciones no tienen porque ser unión compatibles. El resultado del producto cartesiano de $R(A_1, A_2, \ldots, A_n) \times S(B_1, B_2, \ldots, B_m)$ es una relación Q con $n + m$ atributos $Q(A_1, A_2, \ldots, A_n, B_1, B_2, \ldots, B_m)$, en ese orden. La relación resultante Q tiene una tupla por cada combinación de tuplas: una de R y una de S. Por tanto, si R tiene n_R tuplas y S tiene n_S tuplas, $R \times S$ tendrá $n_R \times n_S$ tuplas. Generalmente la operación de producto cartesiano por sí sola no tiene mucho sentido, pero combinada puede tener utilidad. Como se aprecia en la Figura 4.5, el producto cartesiano de DEPARTAMENTO con LOCALIZACIONES DE DEPARTAMENTO no aporta ninguna información, pero si al resultado del producto cartesiano le aplicamos una selección, podemos tener información relevante.

Por ejemplo, si calculamos:

$$R1 \leftarrow Departamento \times Localizaciones\_Dept$$
$$RESULTADO \leftarrow \pi_{Nombre\_Dept,\ Localizacion\_Dept}(\sigma_{Departamento.Num\_Dept=Localizaciones\_Dept.Numero\_Dept}(R1)$$

obtenemos las diferentes localizaciones de un departamento (ver Figura 4.6).

Departamento

Nombre_Dept	Num_Dept	NSS-Jefe	Fech_Ini_Jefe
Dirección	1	1122	12-8-95
Desarrollo	2	5544	25-5-97

Localizaciones_Dept

Número_Dept	Localización_Dept
1	A Coruña
2	Ferrol
2	Lugo
2	Vigo

Departamento X Localizaciones_Dept

Nombre_Dept	Num_Dept	NSS_Jefe	Fech_Ini_Jefe	Número_Dept	Localización_Dept
Dirección	1	1122	12-8-95	1	A Coruña
Dirección	1	1122	12-8-95	2	Ferrol
Dirección	1	1122	12-8-95	2	Lugo
Dirección	1	1122	12-8-95	2	Vigo
Desarrollo	2	5544	25-5-97	1	A Coruña
Desarrollo	2	5544	25-5-97	2	Ferrol
Desarrollo	2	5544	25-5-97	2	Lugo
Desarrollo	2	5544	25-5-97	2	Vigo

Figura 4.5: Ilustración de la operación PRODUCTO CARTESIANO.

Resultado

Nombre_Dept	Localización_Dept
Dirección	A Coruña
Desarrollo	Ferrol
Desarrollo	Lugo
Desarrollo	Vigo

Figura 4.6: La relación RESULTADO.

4.5. Join

La operación JOIN, denotada por $\bowtie$, sirve para combinar tuplas relacionadas de dos relaciones en una sola tupla. Esta operación es muy importante en cualquier base de datos relacional que comprenda más de una relación, porque permite procesar las asociaciones establecidas por las claves externas entre las relaciones. Volviendo al ejemplo de los departamentos y sus localizaciones, antes para poder saber las distintas localizaciones de cada departamento, primero realizábamos un producto cartesiano de las relaciones DEPARTAMENTO y LOCALIZACIONES_DEPT para obtener todas las posibles combinaciones de tuplas de las dos relaciones. Sobre el resultado del producto cartesiano, seleccionamos aquellas tuplas que tienen el mismo valor en los atributos *Num_Dept* de DEPARTAMENTO y *Numero_Dept* de LOCALIZACION_DEPT, puesto que *Numero_Dept* en LOCALIZACION_DEPT es una clave externa que hace referencia a la clave primaria de DEPARTAMENTO (*Num_Dept*). Este mismo efecto se logra aplicando el JOIN. En la Figura 4.7 se puede ver que las dos expresiones de álgebra relacional que aparecen en la figura dan como resultado la misma relación.

De este modo ahorramos una operación, es decir, en el caso de utilizar el producto cartesiano, habría que luego aplicar una selección, pero con el join, sólo es necesaria una operación.

σ Departamento.Num_Dept=Localizaciones_Dept.Numero_Dept (Departamento X Localizaciones_Dept)

Departamento $\bowtie_{\text{Departamento.Num\_Dept=Localizaciones\_Dept.Numero\_Dept}}$ Localizaciones_Dept

Nombre_Dept	Num_Dept	NSS_Jefe	Fech_Ini_Jefe	Número_Dept	Localización_Dept
Dirección	1	1122	12-8-95	1	A Coruña
Desarrollo	2	5544	25-5-97	2	Ferrol
Desarrollo	2	5544	25-5-97	2	Lugo
Desarrollo	2	5544	25-5-97	2	Vigo

Figura 4.7: Ilustración de la operación JOIN.

La forma general de una operación JOIN con dos relaciones $R(A_1, A_2, \ldots, A_n)$ y $S(B_1, B_2, \ldots, B_m)$ es:

$$R \bowtie_{<condicion\ de\ join>} S$$

El resultado del JOIN es una relación Q con $n+m$ atributos $Q(A_1, A_2, \ldots, A_n, B_1, B_2, \ldots, B_m)$, en ese orden. Q tiene una tupla por cada combinación de tuplas (una de R y otra de S) *siempre que la combinación satisfaga la condición de join*. Esta es la principal diferencia entre el PRODUCTO CARTESIANO y el JOIN; en el JOIN sólo aparecen en el resultado combinaciones de tuplas que satisfagan la condición de join. En cambio, en el PRODUCTO CARTESIANO, aparecen *todas* las combinaciones de tuplas. La condición de join se especifica en términos de los atributos de las dos relaciones, R y S, y se evalúa para cada combinación de tuplas de las dos relaciones. Cada combinación que de "verdadero" al evaluarse la condición de join sobre ella aparecerá en la relación resultado Q como *una sola tupla*.

Una condición de join tiene la forma:

$$< condicion > \ AND \ < condicion > \ AND \ \ldots \ AND \ < condicion >$$

donde cada condición tiene la forma $A_i \Theta B_j$, donde A_i es un atributo de R, B_j es un atributo de S, A_i y B_j tienen el mismo dominio y Θ es uno de los operadores de comparación $\{=, <, \leq, >, \geq, \neq\}$. Una operación de JOIN con una condición general de join como ésta se denomina THETA JOIN. Las tuplas cuyos atributos de join sean nulos no aparecen en el resultado.

El JOIN más común implica condiciones de join con comparaciones de igualdad exclusivamente. Un JOIN así en la que el único operador de comparación empleado es $=$, se denomina EQUIJOIN. El ejemplo presentado anteriormente es un EQUIJOIN. Obsérvese que en el resultado de un EQUIJOIN siempre tenemos uno o más pares de atributos con valores idénticos en todas las tuplas (aquellos que especifican la condición de join). Por ejemplo en la Figura 4.7, los valores de los atributos *Num_Dept* y *Número_Dept* son idénticos en todas las tuplas porque se especificó una condición de igualdad para estos dos atributos.

Puesto que uno de cada par de atributos con valores idénticos es superfluo, se ha creado una nueva operación, llamada JOIN NATURAL, para deshacerse de los atributos superfluos en el resultado de un EQUIJOIN. La definición de JOIN NATURAL (denotado por $\bowtie$ sin nada más) exige que los pares de atributos sobre los que el EQUIJOIN establece las condiciones de igualdad tengan el mismo nombre en las dos relaciones. En el siguiente ejemplo primero se renombra el atributo *Numero_Dept* de LOCALIZACIONES_DEPT a *Num_Dept* para posteriormente aplicar un JOIN NATURAL.

$$LOC(Num\_Dept,\ Loc\_Dept) \leftarrow LOCALIZACIONES\_DEPT$$
$$DEPT \leftarrow DEPARTAMENTO \bowtie LOC$$

El atributo *Num_Dept* se denomina *atributo de join*. La Figura **??** muestra el resultado de la expresión anterior.

DEPT

Nombre_Dept	Num_Dept	NSS_Jefe	Fech_Ini_Jefe	Localización_Dept
Dirección	1	1122	12-8-95	A Coruña
Desarrollo	2	5544	25-5-97	Ferrol
Desarrollo	2	5544	25-5-97	Lugo
Desarrollo	2	5544	25-5-97	Vigo

Figura 4.8: Resultado de una operación de JOIN NATURAL.

En la relación DEPT, cada tupla combina una tupla de DEPARTAMENTO y otra de LOCALIZACIONES_DEPT, pero sólo se conserva un *atributo de join*.

En general, el JOIN NATURAL se realiza igualando *todos* los pares de atributos que tienen el mismo nombre en las dos relaciones. Puede haber una lista de atributos de join de cada relación, y cada par correspondiente debe tener el mismo nombre.

4.6. División

La operación DIVISIÓN es útil para un tipo especial de consultas que se presenta a veces en aplicaciones de bases de datos. Un ejemplo es: "obtener los datos de los empleados que trabajan en *todos* los proyectos en los que trabaja Ana Moreira". Para expresar esta consulta con la operación DIVISIÓN, procedemos como sigue. Primero, obtenemos la lista de los números de los proyectos en los que trabaja Ana Moreira, colocando el resultado en la relación intermedia NUMP_ANA:

$$ANA \leftarrow \sigma_{Nombre='AnaMoreira'\ AND\ Apellido1='Moreira'}(EMPLEADO)$$
$$NUMP\_ANA \leftarrow \pi_{Num\_Proy}(TRABAJA\_EN \bowtie_{NSSE=NSS} ANA)$$

Después, creamos una relación intermedia que incluye una tupla <Num_Proy, NSSE> por cada vez que el empleado cuyo número de seguridad social es NSSE trabaja en el proyecto cuyo número es *Num_Proy*:

$$NSS\_NUMP \leftarrow \pi_{Num\_Proy,\ NSSE}(TRABAJA_EN)$$

Por último, aplicamos la operación DIVISIÓN a las dos relaciones, obteniendo los números de seguridad social de los empleados que queremos:

$$NSSS(NSS) \leftarrow NSS\_NUMP \div NUMP\_ANA$$
$$RESULTADO \leftarrow \pi_{Nombre,\ Apellido1}(NSSS \bowtie EMPLEADO)$$

Empleado

Nombre	Apellido1	Apellido2	NSS	F-Nac	Dirección	Sexo	Salario	Supervisor	Num-Dept
José	López	Gómez	1245	21-1-71	Real 8	Varón	34000	5544	2
Ana	Moreira	González	5874	25-7-85	Percebe 1	Mujer	38000	5544	2
María	Pérez	Mosquera	5544	21-8-69	Lechuga 6	Mujer	50000	Nulo	2
Pedro	González	Ruíz	8811	7-7-79	Puerro 10	Varón	25000	1122	1
Luis	Izquierdo	Sánchez	1122	9-9-70	Calcetín 5	Varón	30000	Nulo	1

Trabaja-en

NSSE	Num-Proy	Horas
1245	1	10
1245	3	15
5874	1	20
5544	3	50
5544	1	5
8811	4	10
1122	2	20
5874	3	10

NSS_NUMP

NSSE	Num-Proy
1245	1
1245	3
5874	1
5544	3
5544	1
8811	4
1122	2
5874	3

Nump_Ana

Num-Proy
1
3

NSSS

NSS
1245
5874
5544

Figura 4.9: Ilustración de la operación DIVISIÓN.

En la Figura 4.9 se muestran las relaciones resultado de las operaciones anteriores. En general, la operación DIVISIÓN se aplica a dos relaciones $R(Z) \div S(X)$, donde $X \subseteq Z$. Sea $Y = Z - X$ (y por tanto $Z = X \cup Y$), es decir, Y es el conjunto de atributos de R que non son atributos de S. El resultado de la DIVISIÓN es una relación $T(Y)$ que incluye una tupla t si hay un conjunto de tuplas t_R que aparecen en R, tal que para todas las tuplas t_{Ri} en t_R, $t_{Ri}[Y] = t$ y $\exists\ t_{Rk}[X] = t_{Sj},\ t_{Rk} \subseteq t_R$ para *cada* tupla t_{Sj} en S. Esto significa que, para que una tupla t aparezca en el resultado T de la DIVISIÓN, los valores de t deben aparecer en R en combinación con *todas* las tuplas de S.

La Figura 4.10 ilustra la operación DIVISIÓN donde $X = \{A\},\ Y = \{B\}$ y $Z = \{A, B\}$. Obsérvese que b_1 y b_4 aparecen en R en combinación con las tres tuplas de S, por eso aparecen en la relación resultante T. Todos los demás valores de B en R no aparecen con todas las tuplas de S y no se seleccionan.

La operación DIVISIÓN se puede expresar como una secuencia de operaciones π, $\times$ y $-$, como sigue:

$$T_1 \leftarrow \pi_Y(R)$$
$$T_2 \leftarrow \pi_Y((S \times T_1) - R)$$
$$T \leftarrow T_1 - T_2$$

4.7. Extensiones del álgebra relacional

Después de que Codd introdujera el álgebra relacional como lenguaje de consulta de bases de datos relacionales, numerosos investigadores han propuesto nuevos operadores, en general, necesarios para consultas comunes a bases de datos relacionales.

De entre todas ellas vamos a destacar una extensión que está presente en la mayoría de los SGBD comerciales y es de utilidad, el JOIN EXTERNO.

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

S	A
	a1
	a2
	a3

T	B
	b1
	b4

Figura 4.10: $T \leftarrow R \div S$.

4.7.1. Join Externo

Las operaciones de JOIN antes descritas seleccionan tuplas que satisfacen la condición de join. Las tuplas sin una *tupla relacionada* se eliminan del resultado. Las tuplas que tienen un nulo en los atributos de join también se eliminan. Podemos utilizar el JOIN EXTERIOR cuando queramos conservar en el resultado todas las tuplas que estén en R, en S o en ambas, ya sea que tengan o no tuplas coincidentes en la otra relación. Esto satisface la necesidad de las consultas donde las tuplas de las dos tablas se van a combinar para emparejar las correspondientes filas, pero algunas tuplas se van a perder por no tener valores coincidentes con tuplas de la otra relación. En tales casos, es deseable tener una operación que conserve todas las tuplas tanto si se pueden emparejar como si no.

Por ejemplo, supóngase que deseamos una lista de todos los nombres de empleados y también el nombre de los departamentos que dirigen, *si es el caso de que dirijan un departamento*. Podemos aplicar una operación JOIN EXTERIOR IZQUIERDO, denotado por $\sqsupset\!\!\bowtie$, para obtener el resultado como sigue:

$$TEMP \leftarrow (EMPLEADO \sqsupset\!\!\bowtie_{NSS=NSS\_Jefe} DEPARTAMENTO)$$
$$RESULTADO \leftarrow \pi_{Nombre,\ Apellido1,\ Apellido2,\ Nombre\_Dept}(TEMP)$$

La operación de JOIN EXTERIOR IZQUIERDO conserva todas las tuplas de la primera relación R (o relación de la *izquierda*) en $R \sqsupset\!\!\bowtie S$, si no se encuentra una tupla coincidente en S. Los atributos de S del resultado se “rellenan” con valores nulos. El resultado de estas operaciones se muestra en la figura 4.11.

Una operación similar, el JOIN EXTERIOR DERECHO, denotado por $\bowtie\!\!\sqsubset$, conserva en el resultado de $R \bowtie\!\!\sqsubset S$ todas las tuplas de la segunda relación S (la de la *derecha*). Una tercera operación, JOIN EXTERIOR COMPLETO, denotado por $\sqsupset\!\!\bowtie\!\!\sqsubset$, conserva todas las tuplas de ambas relaciones, izquierda y derecha, cuando no se encuentran tuplas coincidentes, rellenándolas con valores nulos si es necesario.

Resultado

Nombre	Apellido1	Apellido2	Nombre-Dept
José	López	Gómez	Nulo
Ana	Moreira	González	Nulo
María	Pérez	Mosquera	Desarrollo
Pedro	González	Ruíz	Nulo
Luis	Izquierdo	Sánchez	Dirección

Figura 4.11: Operación JOIN EXTERIOR IZQUIERDO.