

# El modelo relacional

El modelo relacional es hoy en día el principal modelo de datos para las aplicaciones comerciales de procesamiento de datos. Ha conseguido esa posición destacada debido a su simplicidad, lo cual facilita el trabajo del programador en comparación con modelos anteriores, como el de red y el jerárquico.

En este capítulo se estudian en primer lugar los fundamentos del modelo relacional. A continuación se describe el álgebra relacional, que se usa para especificar las solicitudes de información. El álgebra relacional no es cómoda de usar, pero sirve de base formal para lenguajes de consultas que sí lo son y que se estudiarán más adelante, incluido el ampliamente usado lenguaje de consultas SQL, el cual se trata con detalle en los Capítulos 3 y 4.

Existe una amplia base teórica para las bases de datos relacionales. En este capítulo se estudia la parte de esa base teórica referida a las consultas. En los Capítulos 6 y 7 se examinarán aspectos de la teoría de las bases de datos relacionales que ayudan en el diseño de esquemas de bases de datos relacionales, mientras que en los Capítulos 13 y 14 se estudian aspectos de la teoría que se refieren al procesamiento eficiente de consultas.

## 2.1 La estructura de las bases de datos relacionales

Una base de datos relacional consiste en un conjunto de **tablas**, a cada una de las cuales se le asigna un nombre exclusivo. Cada fila de la tabla representa una *relación* entre un conjunto de valores. De manera informal, cada tabla es un conjunto de entidades, y cada fila es una entidad, tal y como se estudió en el Capítulo 1. Dado que cada tabla es un conjunto de tales relaciones, hay una fuerte correspondencia entre el concepto de *tabla* y el concepto matemático de *relación*, del que toma su nombre el modelo de datos relacional. A continuación se introduce el concepto de relación.

En este capítulo se usarán varias relaciones diferentes para ilustrar los diversos conceptos subyacentes al modelo de datos relacional. Estas relaciones representan parte de una entidad bancaria. Puede que no se correspondan con el modo en que se pueda estructurar realmente una base de datos bancaria, pero así se simplificará la presentación. Los criterios sobre la adecuación de las estructuras relacionales se estudian con gran detalle en los Capítulos 6 y 7.

### 2.1.1 Estructura básica

Considérese la tabla *cuenta* de la Figura 2.1. Tiene tres cabeceras de columna: *número\_cuenta*, *nombre\_sucursal* y *saldo*. Siguiendo la terminología del modelo relacional, se puede hacer referencia a estas cabeceras como **atributos**. Para cada atributo hay un conjunto de valores permitidos, denominado **dominio** de ese atributo. Para el atributo *nombre\_sucursal*, por ejemplo, el dominio es el conjunto de todos los nombres de sucursal. Supóngase que  $D_1$  denota el conjunto de todos los números de cuenta,  $D_2$  el conjunto de todos los nombres de sucursal y  $D_3$  el conjunto de todos los saldos. Todas las filas de *cuenta* deben consistir en una tupla  $(v_1, v_2, v_3)$ , donde  $v_1$  es un número de cuenta (es decir,  $v_1$  está en

número_cuenta	nombre_sucursal	saldo
C-101	Centro	500
C-102	Navacerrada	400
C-201	Galapagar	900
C-215	Becerril	700
C-217	Galapagar	750
C-222	Moralzarzal	700
C-305	Collado Mediano	350

Figura 2.1 La relación *cuenta*.

el dominio  $D_1$ ,  $v_2$  es un nombre de sucursal ( $v_2$  en  $D_2$ ) y  $v_3$  es un saldo ( $v_3$  en  $D_3$ ). En general, *cuenta* sólo contendrá un subconjunto del conjunto de todas las filas posibles. Por tanto, *cuenta* será un subconjunto de

$$D_1 \times D_2 \times D_3$$

En general, una **tabla** de  $n$  atributos debe ser un subconjunto de

$$D_1 \times D_2 \times \cdots \times D_{n-1} \times D_n$$

Los matemáticos definen las **relaciones** como subconjuntos del producto cartesiano de la lista de dominios. Esta definición se corresponde de manera casi exacta con la definición de *tabla* dada anteriormente. La única diferencia es que aquí se han asignado nombres a los atributos, mientras que los matemáticos sólo usan “nombres” numéricos, usando el entero 1 para denotar el atributo cuyo dominio aparece en primer lugar de la lista de dominios, 2 para el atributo cuyo dominio aparece en segundo lugar, etc. Como las tablas son, esencialmente, relaciones, se usarán los términos matemáticos **relación** y **tupla** en lugar de los términos **tabla** y **fila**. Una **variable tupla** es una variable que representa una tupla; en otras palabras, una variable tupla es una variable cuyo dominio es el conjunto de todas las tuplas.

En la relación *cuenta* de la Figura 2.1 hay siete tuplas. Supóngase que la variable tupla  $t$  hace referencia a la primera tupla de la relación. Se usa la notación  $t[\text{número\_cuenta}]$  para denotar el valor de  $t$  en el atributo *número\_cuenta*. Por tanto,  $t[\text{número\_cuenta}] = \text{“C-101”}$  y  $t[\text{nombre\_sucursal}] = \text{“Centro”}$ . De manera alternativa, se puede escribir  $t[1]$  para denotar el valor de la tupla  $t$  en el primer atributo (*número\_cuenta*),  $t[2]$  para denotar *nombre\_sucursal*, etc. Dado que las relaciones son conjuntos de tuplas, se usa la notación matemática  $t \in r$  para denotar que la tupla  $t$  está en la relación  $r$ .

El orden en que aparecen las tuplas en cada relación es irrelevante, dado que una relación es un *conjunto* de tuplas. Por tanto, no importa si las tuplas de una relación aparecen ordenadas, como en la Figura 2.1, o desordenadas, como en la Figura 2.2; las relaciones de las dos figuras son la misma, ya que las dos contienen el mismo conjunto de tuplas.

Se exige que, para todas las relaciones  $r$ , los dominios de todos los atributos de  $r$  sean atómicos. Un dominio es **atómico** si los elementos del dominio se consideran unidades indivisibles. Por ejemplo, el conjunto de los enteros es un dominio atómico, pero el conjunto de todos los conjuntos de enteros

número_cuenta	nombre_sucursal	saldo
C-101	Centro	500
C-215	Becerril	700
C-102	Navacerrada	400
C-305	Collado Mediano	350
C-201	Galapagar	900
C-222	Moralzarzal	700
C-217	Galapagar	750

Figura 2.2 La relación *cuenta* con las tuplas desordenadas.

es un dominio no atómico. La diferencia es que no se suele considerar que los enteros tengan partes constituyentes, pero sí se considera que los conjuntos de enteros las tienen; por ejemplo, los enteros que forman cada conjunto. Lo importante no es lo que sea el propio dominio, sino la manera en que se usan los elementos del dominio en la base de datos. El dominio de todos los enteros sería no atómico si se considerara que cada entero es una lista ordenada de cifras. En todos los ejemplos se supondrá que los dominios son atómicos. En el Capítulo 9 se estudiarán extensiones al modelo de datos relacional para permitir dominios no atómicos.

Es posible que varios atributos tengan el mismo dominio. Por ejemplo, supóngase la relación *cliente* con los tres atributos *nombre\_cliente*, *calle\_cliente* y *ciudad\_cliente* y una relación *empleado* con el atributo *nombre\_empleado*. Es posible que los atributos *nombre\_cliente* y *nombre\_empleado* tengan el mismo dominio, el conjunto de todos los nombres de persona, que en el nivel físico es el conjunto de todas las cadenas de caracteres. Los dominios de *saldo* y *nombre\_sucursal*, por otra parte, deberían ser distintos. Quizás sea menos evidente si *nombre\_cliente* y *nombre\_sucursal* deberían tener el mismo dominio. En el nivel físico, tanto los nombres de los clientes como los nombres de las sucursales son cadenas de caracteres. Sin embargo, en el nivel lógico puede que se desee que *nombre\_cliente* y *nombre\_sucursal* tengan dominios diferentes.

Un valor de dominio que es miembro de todos los dominios posibles es el valor **nulo**, que indica que el valor es desconocido o no existe. Por ejemplo, supóngase que se incluye el atributo *número\_teléfono* en la relación *cliente*. Puede ocurrir que algún cliente no tenga número de teléfono, o que su número de teléfono no figure en la guía. Entonces habrá que recurrir a los valores nulos para indicar que el valor es desconocido o que no existe. Más adelante se verá que los valores nulos crean algunas dificultades cuando se tiene acceso a la base de datos o se la actualiza y que, por tanto, deben eliminarse si es posible. Se supondrá inicialmente que no hay valores nulos y en el Apartado 2.5 se describirá el efecto de los valores nulos en las diferentes operaciones.

### 2.1.2 Esquema de la base de datos

Cuando se habla de bases de datos se debe diferenciar entre el **esquema de la base de datos**, que es el diseño lógico de la misma, y el **ejemplar de la base de datos**, que es una instantánea de los datos de la misma en un momento dado.

El concepto de relación se corresponde con el concepto de variable de los lenguajes de programación. El concepto de **esquema de la relación** se corresponde con el concepto de definición de tipos de los lenguajes de programación.

Resulta conveniente dar nombre a los esquemas de las relaciones, igual que se dan nombres a las definiciones de los tipos en los lenguajes de programación. Se adopta el convenio de usar nombres en minúsculas para las relaciones y nombres que comiencen por una letra mayúscula para los esquemas de las relaciones. Siguiendo esta notación se usará *Esquema\_cuenta* para denotar el esquema de la relación *cuenta*. Por tanto,

$$Esquema\_cuenta = (\text{número\_cuenta}, \text{nombre\_sucursal}, \text{saldo})$$

Se denota el hecho de que *cuenta* es una relación de *Esquema\_cuenta* mediante

$$cuenta(Esquema\_cuenta)$$

En general, los esquemas de las relaciones consisten en una lista de los atributos y de sus dominios correspondientes. La definición exacta del dominio de cada atributo no será relevante hasta que se estudie el lenguaje SQL en los Capítulos 3 y 4.

El concepto de **ejemplar de la relación** se corresponde con el concepto de valor de una variable en los lenguajes de programación. El valor de una variable dada puede cambiar con el tiempo; de manera parecida, el contenido del ejemplar de una relación puede cambiar con el tiempo cuando la relación se actualiza. Sin embargo, se suele decir simplemente “relación” cuando realmente se quiere decir “ejemplar de la relación”.

Como ejemplo de ejemplar de una relación, considérese la relación *sucursal* de la Figura 2.3. El esquema de esa relación es

<i>nombre_sucursal</i>	<i>ciudad_sucursal</i>	<i>activos</i>
Becerril	Aluche	400.000
Centro	Arganzuela	9.000.000
Collado Mediano	Aluche	8.000.000
Galapagar	Arganzuela	7.100.000
Moralzarzal	La Granja	2.100.000
Navacerrada	Aluche	1.700.000
Navas de la Asunción	Alcalá de Henares	300.000
Segovia	Cerceda	3.700.000

**Figura 2.3** La relación *sucursal*.

$$\text{Esquema\_sucursal} = (\text{nombre\_sucursal}, \text{ciudad\_sucursal}, \text{activos})$$

Obsérvese que el atributo *nombre\_sucursal* aparece tanto en *Esquema\_sucursal* como en *Esquema\_cuenta*. Esta duplicidad no es una coincidencia. Más bien, usar atributos comunes en los esquemas de las relaciones es una manera de relacionar las tuplas de relaciones diferentes. Por ejemplo, supóngase que se desea obtener información sobre todas las cuentas abiertas en sucursales ubicadas en Arganzuela. Primero se busca en la relación *sucursal* para encontrar los nombres de todas las sucursales situadas en Arganzuela. A continuación y para cada una de ellas, se examina la relación *cuenta* para encontrar la información sobre las cuentas abiertas en esa sucursal.

Siguiendo con el ejemplo bancario, se necesita una relación que describa información sobre los clientes. El esquema de la relación es:

$$\text{Esquema\_cliente} = (\text{nombre\_cliente}, \text{calle\_cliente}, \text{ciudad\_cliente})$$

La Figura 2.4 muestra un ejemplo de la relación *cliente* (*Esquema\_cliente*). Obsérvese que se ha omitido el atributo *id\_cliente*, que se usó en el Capítulo 1, ya que se considerarán esquemas de relación más pequeños en nuestro ejemplo de una base de datos bancaria. Se da por supuesto que el nombre de cliente identifica unívocamente a cada cliente—obviamente, puede que esto no sea cierto en el mundo real, pero la suposición hace los ejemplos más sencillos de entender. En una base de datos del mundo real, *id\_cliente* (que podría ser *número\_seguridad\_social* o un identificador generado por el banco) serviría para identificar unívocamente a los clientes.

También se necesita una relación que describa la asociación entre los clientes y las cuentas. El esquema de la relación que describe esta asociación es:

$$\text{Esquema\_impositor} = (\text{nombre\_cliente}, \text{número\_cuenta})$$

<i>nombre_cliente</i>	<i>calle_cliente</i>	<i>ciudad_cliente</i>
Abril	Preciados	Valsaín
Amo	Embajadores	Arganzuela
Badorrey	Delicias	Valsaín
Fernández	Jazmín	León
Gómez	Carretas	Cerceda
González	Arenal	La Granja
López	Mayor	Peguerinos
Pérez	Carretas	Cerceda
Rodríguez	Yeserías	Cádiz
Rupérez	Ramblas	León
Santos	Mayor	Peguerinos
Valdivieso	Goya	Vigo

**Figura 2.4** La relación *cliente*.

<i>nombre_cliente</i>	<i>número_cuenta</i>
Abril	C-305
Gómez	C-215
González	C-101
González	C-201
López	C-102
Rupérez	C-222
Santos	C-217

**Figura 2.5** La relación *impositor*.

La Figura 2.5 muestra un ejemplo de la relación *impositor* (*Esquema\_impositor*).

Puede parecer que para este ejemplo bancario se podría tener sólo un esquema de relación, en vez de tener varios. Es decir, puede resultar más sencillo para el usuario pensar en términos de un único esquema de relación, en lugar de en varios esquemas. Supóngase que sólo se usara una relación para el ejemplo, con el esquema

(*nombre\_sucursal, ciudad\_sucursal, activos, nombre\_cliente, calle\_cliente*  
*ciudad\_cliente, número\_cuenta, saldo*)

Obsérvese que, si un cliente tiene varias cuentas, hay que repetir su dirección una vez por cada cuenta. Es decir, hay que repetir varias veces parte de la información. Esta repetición malgasta espacio, pero se evita mediante el empleo de varias relaciones mostradas anteriormente.

Además, si una sucursal no tiene ninguna cuenta (por ejemplo, una sucursal recién abierta que todavía no tenga clientes), no se puede construir una tupla completa en la relación única anterior, dado que no hay todavía ningún dato disponible referente a *cliente* ni a *cuenta*. Para representar las tuplas incompletas hay que usar valores *nulos* que indiquen que ese valor es desconocido o no existe. Por tanto, en el ejemplo presente, los valores de *nombre\_cliente*, *calle\_cliente*, etc., deben ser nulos. Al emplear varias relaciones se puede representar la información de las sucursales del banco sin clientes sin necesidad de valores nulos. Se usa simplemente una tupla en *Esquema\_sucursal* para representar la información de la sucursal, y sólo se crean tuplas en los otros esquemas cuando esté disponible la información correspondiente.

En el Capítulo 7 se estudiarán los criterios para decidir cuándo un conjunto de esquemas de relaciones es más adecuado que otro en términos de repetición de la información y de la existencia de valores nulos. Por ahora se supondrá que los esquemas de las relaciones vienen dados de antemano.

Se incluyen dos relaciones más para describir los datos de los préstamos concedidos en las diferentes sucursales del banco:

*Esquema\_préstamo* = (*número\_préstamo, nombre\_sucursal, importe*)  
*Esquema\_prestatario* = (*nombre\_cliente, número\_préstamo*)

Las Figuras 2.6 y 2.7, respectivamente, muestran las relaciones de ejemplo *préstamo* (*Esquema\_préstamo*) y *prestatario* (*Esquema\_prestatario*).

Los esquemas de relación se corresponden con el conjunto de tablas que podrían generarse con el método descrito en el Apartado 1.6. Obsérvese que la relación *cliente* puede contener información sobre clientes que no tengan ni cuenta ni préstamo en el banco. La entidad bancaria aquí descrita servirá como ejemplo principal en este capítulo. Cuando sea necesario, habrá que introducir más esquemas de relaciones para ilustrar casos concretos.

### 2.1.3 Claves

Es necesario disponer de un modo de especificar la manera en que las tuplas de una relación dada se distinguen entre sí. Esto se expresa en términos de sus atributos. Es decir, los valores de los atributos de una tupla deben ser tales que puedan *identificarla unívocamente*. En otras palabras, no

número_prestamo	nombre_sucursal	importe
P-11	Collado Mediano	900
P-14	Centro	1.500
P-15	Navacerrada	1.500
P-16	Navacerrada	1.300
P-17	Centro	1.000
P-23	Moralzarzal	2.000
P-93	Becerril	500

Figura 2.6 La relación *préstamo*.

se permite que dos tuplas de una misma relación tengan exactamente los mismos valores en todos sus atributos.

Una **superclave** es un conjunto de uno o varios atributos que, considerados conjuntamente, permiten identificar de manera unívoca una tupla de la relación. Por ejemplo, el atributo *id\_cliente* de la relación *cliente* es suficiente para distinguir una tupla *cliente* de otra. Por tanto, *id\_cliente* es una superclave. De manera parecida, la combinación de *nombre\_cliente* e *id\_cliente* constituye una superclave para la relación *cliente*. El atributo *nombre\_cliente* de *cliente* no es una superclave, ya que es posible que varias personas se llamen igual.

El concepto de superclave no es suficiente para nuestros propósitos, ya que, como se ha podido ver, las superclaves pueden contener atributos innecesarios. Si *C* es una superclave, entonces también lo es cualquier superconjunto de *C*. A menudo resultan interesantes superclaves para las que ninguno de sus subconjuntos constituya una superclave. Esas superclaves mínimas se denominan **claves candidatas**.

Es posible que varios conjuntos diferentes de atributos puedan ejercer como claves candidatas. Supóngase que una combinación de *nombre\_cliente* y de *calle\_cliente* sea suficiente para distinguir entre los miembros de la relación *cliente*. Entonces, tanto {*id\_cliente*} como {*nombre\_cliente*, *calle\_cliente*} son claves candidatas. Aunque los atributos *id\_cliente* y *nombre\_cliente* en conjunto pueden diferenciar las tuplas *cliente*, su combinación no forma una clave candidata, ya que el atributo *id\_cliente* por sí solo ya lo es.

Se usará el término **clave primaria** para denotar una clave candidata que ha elegido el diseñador de la base de datos como medio principal para la identificación de las tuplas de una relación. Las claves (sean primarias, candidatas o superclaves) son propiedades de toda la relación, no de cada una de las tuplas. Ninguna pareja de tuplas de la relación puede tener simultáneamente el mismo valor de los atributos de la clave. La selección de una clave representa una restricción de la empresa del mundo real que se está modelando.

Las claves candidatas deben escogerse con cuidado. Como se ha indicado, el nombre de una persona evidentemente no es suficiente, ya que puede haber mucha gente con el mismo nombre. En Estados Unidos el atributo número de la seguridad social de cada persona sería clave candidata. Dado que los residentes extranjeros no suelen tener número de la seguridad social, las empresas internacionales deben generar sus propios identificadores unívocos. Una alternativa es usar como clave alguna combinación exclusiva de otros atributos.

nombre_cliente	número_prestamo
Fernández	P-16
Gómez	P-11
Gómez	P-23
López	P-15
Pérez	P-93
Santos	P-17
Sotoca	P-14
Valdivieso	P-17

Figura 2.7 La relación *prestatario*.

La clave primaria debe escogerse de manera que los valores de sus atributos no se modifiquen nunca, o muy rara vez. Por ejemplo, el campo domicilio de una persona no debe formar parte de la clave primaria, ya que es probable que se modifique. Por otra parte, está garantizado que los números de la seguridad social no cambian nunca. Los identificadores exclusivos generados por las empresas no suelen cambiar, salvo si se produce una fusión entre dos de ellas; en ese caso, puede que el mismo identificador haya sido emitido por ambas empresas, y puede ser necesaria una reasignación de identificadores para garantizar que sean únicos.

Formalmente, sea  $R$  el esquema de una relación. Si se dice que un subconjunto  $C$  de  $R$  es una *superclave* de  $R$ , se restringe la consideración a las relaciones  $r(R)$  en las que no hay dos tuplas diferentes que tengan los mismos valores en todos los atributos de  $C$ . Es decir, si  $t_1$  y  $t_2$  están en  $r$  y  $t_1 \neq t_2$ , entonces  $t_1[C] \neq t_2[C]$ .

El esquema de una relación, por ejemplo  $r_1$ , puede incluir entre sus atributos la clave primaria de otro esquema de relación, por ejemplo  $r_2$ . Este atributo se denomina **clave externa** de  $r_1$ , que hace referencia a  $r_2$ . La relación  $r_1$  también se denomina **relación referenciante** de la dependencia de clave externa, y  $r_2$  se denomina **relación referenciada** de la clave externa. Por ejemplo, el atributo *nombre\_sucursal* de *Esquema\_cuenta* es una clave externa de *Esquema\_cuenta* que hace referencia a *Esquema\_sucursal*, ya que *nombre\_sucursal* es la clave primaria de *Esquema\_sucursal*. En cualquier ejemplar de la base de datos, dada cualquier tupla, por ejemplo  $t_a$ , de la relación *cuenta*, debe haber alguna tupla, por ejemplo  $t_b$ , en la relación *sucursal* tal que el valor del atributo *nombre\_sucursal* de  $t_a$  sea el mismo que el valor de la clave primaria de  $t_b$  *nombre\_sucursal*.

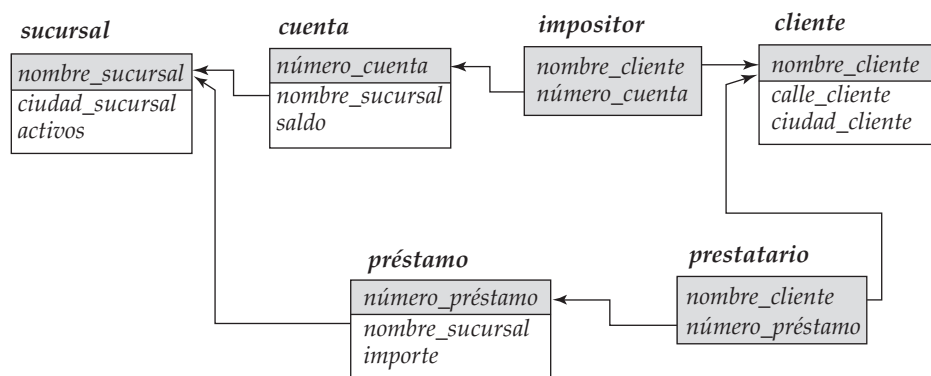
Es costumbre relacionar los atributos de la clave primaria de un esquema de relación antes que el resto de los atributos; por ejemplo, el atributo *nombre\_sucursal* de *Esquema\_sucursal* se relaciona en primer lugar, ya que es la clave primaria.

El esquema de la base de datos, junto con las dependencias de clave primaria y externa, se puede mostrar gráficamente mediante **diagramas de esquema**. La Figura 2.8 muestra el diagrama de esquema del ejemplo bancario. Cada relación aparece como un cuadro con los atributos relacionados en su interior y el nombre de la relación sobre él. Si hay atributos de clave primaria, una línea horizontal cruza el cuadro con los atributos de clave primaria por encima de ella y sobre fondo gris. Las dependencias de clave externa aparecen como flechas desde los atributos de clave externa de la relación referenciante a la clave primaria de la relación referenciada.

Muchos sistemas de bases de datos proporcionan herramientas de diseño con una interfaz gráfica de usuario para la creación de los diagramas de esquema.

#### 2.1.4 Lenguajes de consultas

Un **lenguaje de consultas** es un lenguaje en el que los usuarios solicitan información de la base de datos. Estos lenguajes suelen ser de un nivel superior que el de los lenguajes de programación habituales. Los lenguajes de consultas pueden clasificarse como procedimentales o no procedimentales. En los **lenguajes procedimentales** el usuario indica al sistema que lleve a cabo una serie de operaciones en la base



**Figura 2.8** Diagrama del esquema de la entidad bancaria.



de datos para calcular el resultado deseado. En los **lenguajes no procedimentales** el usuario describe la información deseada sin dar un procedimiento concreto para obtener esa información.

La mayor parte de los sistemas comerciales de bases de datos relacionales ofrecen un lenguaje de consultas que incluye elementos de los enfoques procedimental y no procedimental. Se estudiará el muy usado lenguaje de consultas SQL en los Capítulos 3 y 4. El Capítulo 5 trata los lenguajes de consultas QBE y Datalog; este último es un lenguaje de consultas parecido al lenguaje de programación Prolog.

Existen varios lenguajes de consultas “puros”: el álgebra relacional es procedimental, mientras que el cálculo relacional de tuplas y el cálculo relacional de dominios no lo son. Estos lenguajes de consultas son rígidos y formales, y carecen del “azúcar sintáctico” de los lenguajes comerciales, pero ilustran las técnicas fundamentales para la extracción de datos de las bases de datos.

En este capítulo se examina con gran detalle el lenguaje del álgebra relacional (en el Capítulo 5 se tratan los lenguajes del cálculo relacional de tuplas y del cálculo relacional de dominios). El álgebra relacional consiste en un conjunto de operaciones que toman una o dos relaciones como entrada y generan otra relación nueva como resultado.

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.

Inicialmente sólo se estudiarán las consultas. Sin embargo, un lenguaje de manipulación de datos completo no sólo incluye un lenguaje de consultas, sino también un lenguaje para la modificación de las bases de datos. Este tipo de lenguajes incluye comandos para insertar y borrar tuplas, así como para modificar partes de las tuplas existentes. Las modificaciones de las bases de datos se examinarán después de completar la discusión sobre las consultas.

## 2.2 Operaciones fundamentales del álgebra relacional

Las operaciones selección, proyección y renombramiento se denominan operaciones *unarias* porque operan sobre una sola relación. Las otras tres operaciones operan sobre pares de relaciones y se denominan, por tanto, operaciones *binarias*.

### 2.2.1 Operación selección

La operación **selección** selecciona tuplas que satisfacen un predicado dado. Se usa la letra griega sigma minúscula ( $\sigma$ ) para denotar la selección. El predicado aparece como subíndice de  $\sigma$ . La relación de argumentos se da entre paréntesis a continuación de  $\sigma$ . Por tanto, para seleccionar las tuplas de la relación *préstamo* en que la sucursal es “Navacerrada” se escribe

$$\sigma_{\text{nombre\_sucursal} = \text{“Navacerrada”}} (\text{préstamo})$$

Si la relación *préstamo* es como se muestra en la Figura 2.6, la relación que resulta de la consulta anterior es como aparece en la Figura 2.9.

número_préstamo	nombre_sucursal	importe
P-15	Navacerrada	1.500
P-16	Navacerrada	1.300

**Figura 2.9** Resultado de  $\sigma_{\text{nombre\_sucursal} = \text{“Navacerrada”}} (\text{préstamo})$ .



Se pueden buscar todas las tuplas en las que el importe prestado sea mayor que 1.200 € escribiendo la siguiente consulta:

$$\sigma_{\text{importe} > 1200}(\text{préstamo})$$

En general, se permiten las comparaciones que usan  $=$ ,  $\neq$ ,  $<$ ,  $\leq$ ,  $>$  o  $\geq$  en el predicado de selección. Además, se pueden combinar varios predicados en uno mayor usando las conectivas *y* ( $\wedge$ ), *o* ( $\vee$ ) y *no* ( $\neg$ ). Por tanto, para encontrar las tuplas correspondientes a préstamos de más de 1.200 € concedidos por la sucursal de Navacerrada, se escribe

$$\sigma_{\text{nombre\_sucursal} = \text{"Navacerrada"} \wedge \text{importe} > 1200}(\text{préstamo})$$

El predicado de selección puede incluir comparaciones entre dos atributos. Para ilustrarlo, considérese la relación *responsable\_préstamo*, que consta de tres atributos: *nombre\_cliente*, *nombre\_responsable* y *número\_préstamo*, que especifica que un empleado concreto es el responsable del préstamo concedido a un cliente. Para hallar todos los clientes que se llaman igual que su responsable de préstamos se puede escribir

$$\sigma_{\text{nombre\_cliente} = \text{nombre\_responsable}}(\text{responsable\_préstamo})$$

### 2.2.2 Operación proyección

Supóngase que se desea obtener una relación de todos los números e importes de los préstamos, pero sin los nombres de las sucursales. La operación **proyección** permite obtener esa relación. La operación proyección es una operación unaria que devuelve su relación de argumentos, excluyendo algunos argumentos. Dado que las relaciones son conjuntos, se eliminan todas las filas duplicadas. La proyección se denota por la letra griega mayúscula pi ( $\Pi$ ). Se crea una lista de los atributos que se desea que aparezcan en el resultado como subíndice de  $\Pi$ . Su único argumento, una relación, se escribe a continuación entre paréntesis. La consulta para crear una lista de todos los números e importes de los préstamos puede escribirse como

$$\Pi_{\text{número\_préstamo}, \text{importe}}(\text{préstamo})$$

La Figura 2.10 muestra la relación que resulta de esta consulta.

### 2.2.3 Composición de operaciones relacionales

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

$$\Pi_{\text{nombre\_cliente}}(\sigma_{\text{ciudad\_cliente} = \text{"Peguerinos"}}(\text{cliente}))$$

Téngase en cuenta que, en vez de dar el nombre de una relación como argumento de la operación proyección, se da una expresión cuya evaluación es una relación.

En general, dado que el resultado de las operaciones del álgebra relacional es del mismo tipo (relación) que los datos de entrada, las operaciones del álgebra relacional pueden componerse para formar

<i>número_préstamo</i>	<i>importe</i>
P-11	900
P-14	1.500
P-15	1.500
P-16	1.300
P-17	1.000
P-23	2.000
P-93	500

**Figura 2.10** Números e importes de los préstamos.

una **expresión del álgebra relacional**. Componer operaciones del álgebra relacional para formar expresiones del álgebra relacional es igual que componer operaciones aritméticas (como  $+$ ,  $-$ ,  $*$  y  $\div$ ) para formar expresiones aritméticas. La definición formal de las expresiones de álgebra relacional se estudia en el Apartado 2.2.8.

### 2.2.4 Operación unión

Considérese una consulta para determinar el nombre de todos los clientes del banco que tienen una cuenta, un préstamo o ambas cosas. Obsérvese que la relación *cliente* no contiene esa información, dado que los clientes no necesitan tener ni cuenta ni préstamo en el banco. Para contestar a esta consulta hace falta la información de las relaciones *impositor* (Figura 2.5) y *prestatario* (Figura 2.7). Para determinar los nombres de todos los clientes con préstamos en el banco con las operaciones estudiadas se escribe:

$$\Pi_{\text{nombre\_cliente}}(\text{prestatarario})$$

Igualmente, para determinar el nombre de todos los clientes con cuenta en el banco se escribe:

$$\Pi_{\text{nombre\_cliente}}(\text{impositor})$$

Para contestar a la consulta es necesaria la **unión** de estos dos conjuntos; es decir, hacen falta todos los nombres de clientes que aparecen en alguna de las dos relaciones o en ambas. Estos datos se pueden averiguar mediante la operación binaria unión, denotada, como en la teoría de conjuntos, por  $\cup$ . Por tanto, la expresión buscada es:

$$\Pi_{\text{nombre\_cliente}}(\text{prestatarario}) \cup \Pi_{\text{nombre\_cliente}}(\text{impositor})$$

La relación resultante de esta consulta aparece en la Figura 2.11. Téngase en cuenta que en el resultado hay diez tuplas, aunque haya siete prestatarios y seis impositores distintos. Esta discrepancia aparente se debe a que Gómez, López y Santos son a la vez prestatarios e impositores. Dado que las relaciones son conjuntos, se eliminan los valores duplicados.

Obsérvese que en este ejemplo se toma la unión de dos conjuntos, ambos consistentes en valores de *nombre\_cliente*. En general, se debe asegurar que las uniones se realicen entre relaciones *compatibles*. Por ejemplo, no tendría sentido realizar la unión de las relaciones *préstamo* y *prestatarario*. La primera es una relación con tres atributos, la segunda sólo tiene dos. Más aún, considérese la unión de un conjunto de nombres de clientes y de un conjunto de ciudades. Una unión así no tendría sentido en la mayor parte de los casos. Por tanto, para que la operación unión  $r \cup s$  sea válida hay que exigir que se cumplan dos condiciones:

1. Las relaciones  $r$  y  $s$  deben ser de la misma aridad. Es decir, deben tener el mismo número de atributos.
2. Los dominios de los atributos  $i$ -ésimos de  $r$  y de  $s$  deben ser iguales para todo  $i$ .

<i>nombre_cliente</i>
Abril
Fernández
Gómez
González
López
Pérez
Rupérez
Santos
Sotoca
Valdivieso

**Figura 2.11** Nombre de todos los clientes que tienen un préstamo o una cuenta.

<i>nombre_cliente</i>
Abril
González
Rupérez

**Figura 2.12** Clientes con cuenta abierta pero sin préstamo concedido.

Téngase en cuenta que  $r$  y  $s$  pueden ser, en general, relaciones de la base de datos o relaciones temporales resultado de expresiones del álgebra relacional.

### 2.2.5 Operación diferencia de conjuntos

La operación **diferencia de conjuntos**, denotada por  $-$ , permite hallar las tuplas que están en una relación pero no en la otra. La expresión  $r - s$  da como resultado una relación que contiene las tuplas que están en  $r$  pero no en  $s$ .

Se pueden buscar todos los clientes del banco que tengan abierta una cuenta pero no tengan concedido ningún préstamo escribiendo

$$\Pi_{\text{nombre\_cliente}}(\text{impositor}) - \Pi_{\text{nombre\_cliente}}(\text{prestatario})$$

La relación resultante de esta consulta aparece en la Figura 2.12.

Como en el caso de la operación unión, hay que asegurarse de que las diferencias de conjuntos se realicen entre relaciones *compatibles*. Por tanto, para que una operación diferencia de conjuntos  $r - s$  sea válida se exige que las relaciones  $r$  y  $s$  sean de la misma aridad y que los dominios de los atributos  $i$ -ésimos de  $r$  y de  $s$  sean iguales.

### 2.2.6 Operación producto cartesiano

La operación **producto cartesiano**, denotada por un aspa ( $\times$ ), permite combinar información de cualesquiera dos relaciones. El producto cartesiano de las relaciones  $r_1$  y  $r_2$  se escribe  $r_1 \times r_2$ .

Recuérdese que las relaciones se definen como subconjuntos del producto cartesiano de un conjunto de dominios. A partir de esa definición ya se debe tener una intuición sobre la definición de la operación producto cartesiano. Sin embargo, dado que el mismo nombre de atributo puede aparecer tanto en  $r_1$  como en  $r_2$ , es necesario crear un convenio de denominación para distinguir unos atributos de otros. En este caso se realiza adjuntando al atributo el nombre de la relación de la que proviene originalmente. Por ejemplo, el esquema de relación de  $r = \text{prestatario} \times \text{préstamo}$  es

$$(\text{prestatario.nombre\_cliente}, \text{prestatario.número\_préstamo}, \\ \text{préstamo.número\_préstamo}, \text{préstamo.nombre\_sucursal}, \text{préstamo.importe})$$

Con este esquema se puede distinguir entre  $\text{prestatario.número\_préstamo}$  y  $\text{préstamo.número\_préstamo}$ . Para los atributos que sólo aparecen en uno de los dos esquemas se suele omitir el prefijo con el nombre de la relación. Esta simplificación no genera ambigüedad alguna. Por tanto, se puede escribir el esquema de la relación  $r$  como

$$(\text{nombre\_cliente}, \text{prestatario.número\_préstamo}, \\ \text{préstamo.número\_préstamo}, \text{nombre\_sucursal}, \text{importe})$$

Este convenio de denominaciones exige que las relaciones que sean argumentos de la operación producto cartesiano tengan nombres diferentes. Esta exigencia causa problemas en algunos casos, como cuando se desea calcular el producto cartesiano de una relación consigo misma. Se produce un problema parecido si se usa el resultado de una expresión del álgebra relacional en un producto cartesiano, dado que hará falta un nombre de relación para poder hacer referencia a sus atributos. En el Apartado 2.2.7 se verá la manera de evitar estos problemas mediante la operación renombramiento.

Ahora que se conoce el esquema de relación de  $r = \text{prestatario} \times \text{préstamo}$  es necesario hallar las tuplas que aparecerán en  $r$ . Como es posible imaginar, se crea una tupla de  $r$  a partir de cada par de tuplas

<i>nombre_cliente</i>	<i>prestatario. número_prestamo</i>	<i>préstamo. número_prestamo</i>	<i>nombre_sucursal</i>	<i>importe</i>
Fernández	P-16	P-11	Collado Mediano	900
Fernández	P-16	P-14	Centro	1.500
Fernández	P-16	P-15	Navacerrada	1.500
Fernández	P-16	P-16	Navacerrada	1.300
Fernández	P-16	P-17	Centro	1.000
Fernández	P-16	P-23	Moralzarzal	2.000
Fernández	P-16	P-93	Becerril	500
Gómez	P-11	P-11	Collado Mediano	900
Gómez	P-11	P-14	Centro	1.500
Gómez	P-11	P-15	Navacerrada	1.500
Gómez	P-11	P-16	Navacerrada	1.300
Gómez	P-11	P-17	Centro	1.000
Gómez	P-11	P-23	Moralzarzal	2.000
Gómez	P-11	P-93	Becerril	500
Gómez	P-23	P-11	Collado Mediano	900
Gómez	P-23	P-14	Centro	1.500
Gómez	P-23	P-15	Navacerrada	1.500
Gómez	P-23	P-16	Navacerrada	1.300
Gómez	P-23	P-17	Centro	1.000
Gómez	P-23	P-23	Moralzarzal	2.000
Gómez	P-23	P-93	Becerril	500
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...
Sotoca	P-14	P-11	Collado Mediano	900
Sotoca	P-14	P-14	Centro	1.500
Sotoca	P-14	P-15	Navacerrada	1.500
Sotoca	P-14	P-16	Navacerrada	1.300
Sotoca	P-14	P-17	Centro	1.000
Sotoca	P-14	P-23	Moralzarzal	2.000
Sotoca	P-14	P-93	Becerril	500
Valdivieso	P-17	P-11	Collado Mediano	900
Valdivieso	P-17	P-14	Centro	1.500
Valdivieso	P-17	P-15	Navacerrada	1.500
Valdivieso	P-17	P-16	Navacerrada	1.300
Valdivieso	P-17	P-17	Centro	1.000
Valdivieso	P-17	P-23	Moralzarzal	2.000
Valdivieso	P-17	P-93	Becerril	500

**Figura 2.13** Resultado de *prestatario*  $\times$  *préstamo*.

posible: una de la relación *prestatario* y otra de *préstamo*. Por tanto,  $r$  es una relación de gran tamaño, como se puede ver en la Figura 2.13, donde sólo se ha incluido una parte de las tuplas que constituyen  $r$ .

Supóngase que se tienen  $n_1$  tuplas de *prestatario* y  $n_2$  tuplas de *préstamo*. Por tanto, hay  $n_1 * n_2$  maneras de escoger un par de tuplas—una tupla de cada relación; por lo que hay  $n_1 * n_2$  tuplas en  $r$ . En concreto, obsérvese que para algunas tuplas  $t$  de  $r$  puede ocurrir que  $t[\text{prestatario.número\_préstamo}] \neq t[\text{préstamo.número\_préstamo}]$ .

En general, si se tienen las relaciones  $r_1(R_1)$  y  $r_2(R_2)$ ,  $r_1 \times r_2$  es una relación cuyo esquema es la concatenación de  $R_1$  y de  $R_2$ . La relación  $R$  contiene todas las tuplas  $t$  para las que hay unas tuplas  $t_1$  en  $r_1$  y  $t_2$  en  $r_2$  para las que  $t[R_1] = t_1[R_1]$  y  $t[R_2] = t_2[R_2]$ .

<i>nombre_cliente</i>	<i>prestatario.</i> <i>número_prestamo</i>	<i>préstamo.</i> <i>número_prestamo</i>	<i>nombre_sucursal</i>	<i>importe</i>
Fernández	P-16	P-15	Navacerrada	1.500
Fernández	P-16	P-16	Navacerrada	1.300
Gómez	P-11	P-15	Navacerrada	1.500
Gómez	P-11	P-16	Navacerrada	1.300
Gómez	P-23	P-15	Navacerrada	1.500
Gómez	P-23	P-16	Navacerrada	1.300
López	P-15	P-15	Navacerrada	1.500
López	P-15	P-16	Navacerrada	1.300
Pérez	P-93	P-15	Navacerrada	1.500
Pérez	P-93	P-16	Navacerrada	1.300
Santos	P-17	P-15	Navacerrada	1.500
Santos	P-17	P-16	Navacerrada	1.300
Sotoca	P-14	P-15	Navacerrada	1.500
Sotoca	P-14	P-16	Navacerrada	1.300
Valdivieso	P-17	P-15	Navacerrada	1.500
Valdivieso	P-17	P-16	Navacerrada	1.300

**Figura 2.14** Resultado de  $\sigma_{\text{nombre\_sucursal} = \text{"Navacerrada"}}(\text{prestatario} \times \text{préstamo})$ .

Supóngase que se desea determinar el nombre de todos los clientes que tienen concedido un préstamo en la sucursal de Navacerrada. Se necesita para ello información de las relaciones *préstamo* y *prestatario*. Si se escribe

$$\sigma_{\text{nombre\_sucursal} = \text{"Navacerrada"}}(\text{prestatario} \times \text{préstamo})$$

entonces el resultado es la relación mostrada en la Figura 2.14. Se tiene una relación que sólo atañe a la sucursal de Navacerrada. Sin embargo, la columna *nombre\_cliente* puede contener clientes que no tengan concedido ningún préstamo en la sucursal de Navacerrada. (Si no se ve el motivo por el que esto es cierto, recuérdese que el producto cartesiano toma todos los emparejamientos posibles de cada tupla de *prestatario* con cada tupla de *préstamo*).

Dado que la operación producto cartesiano asocia *todas* las tuplas de *préstamo* con todas las tuplas de *prestatario*, se sabe que, si un cliente tiene concedido un préstamo en la sucursal de Navacerrada, hay alguna tupla de *prestatario*  $\times$  *préstamo* que contiene su nombre y que *prestatario.número\_prestamo* = *préstamo.número\_prestamo*. Por tanto, si se escribe

$$\sigma_{\text{prestatario.número\_prestamo} = \text{préstamo.número\_prestamo}}(\sigma_{\text{nombre\_sucursal} = \text{"Navacerrada"}}(\text{prestatario} \times \text{préstamo}))$$

sólo se obtienen las tuplas de *prestatario*  $\times$  *préstamo* que corresponden a los clientes que tienen concedido un préstamo en la sucursal de Navacerrada.

Finalmente, dado que sólo se desea obtener *nombre\_cliente*, se realiza una proyección:

$$\Pi_{\text{nombre\_cliente}}(\sigma_{\text{prestatario.número\_prestamo} = \text{préstamo.número\_prestamo}}(\sigma_{\text{nombre\_sucursal} = \text{"Navacerrada"}}(\text{prestatario} \times \text{préstamo})))$$

El resultado de esta expresión, mostrada en la Figura 2.15, es la respuesta correcta a la consulta formulada.

### 2.2.7 Operación renombramiento

A diferencia de las relaciones de la base de datos, los resultados de las expresiones de álgebra relacional no tienen un nombre que se pueda usar para referirse a ellas. Resulta útil poder ponerles nombre; la operación **renombramiento**, denotada por la letra griega  $\rho$  minúscula ( $\rho$ ), permite hacerlo. Dada una

<i>nombre_cliente</i>
Fernández
López

**Figura 2.15** Resultado de  $\Pi_{\text{nombre\_cliente}}$ 

$$(\sigma_{\text{prestatario.número\_préstamo} = \text{préstamo.número\_préstamo}} (\sigma_{\text{nombre\_sucursal} = \text{"Navacerrada"}} (\text{prestatario} \times \text{préstamo}))).$$

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  $A_1, A_2, \dots, A_n$ .

Para ilustrar el renombramiento de relaciones, considérese la consulta “Buscar el saldo máximo de las cuentas del banco”. La estrategia empleada para obtener el resultado es (1) calcular en primer lugar una relación temporal consistente en los saldos que *no* son el máximo y (2) realizar la diferencia entre la relación  $\Pi_{\text{saldo}}(\text{cuenta})$  y la relación temporal recién calculada para obtener el resultado.

Paso 1: para calcular la relación intermedia hay que comparar los valores de los saldos de todas las cuentas. Esta comparación se hará calculando el producto cartesiano  $\text{cuenta} \times \text{cuenta}$  y formando una selección para comparar el valor de cualesquiera dos saldos que aparezcan en una tupla. En primer lugar hay que crear un mecanismo para distinguir entre los dos atributos *saldo*. Se usará la operación renombramiento para cambiar el nombre de una referencia a la relación *cuenta*; de este modo se puede hacer referencia dos veces a la relación sin ambigüedad alguna.

La relación temporal que se compone de los saldos que no son el máximo puede escribirse ahora como:

$$\Pi_{\text{cuenta.saldo}} (\sigma_{\text{cuenta.saldo} < d.\text{saldo}} (\text{cuenta} \times \rho_d(\text{cuenta})))$$

Esta expresión proporciona los saldos de la relación *cuenta* para los que aparece un saldo mayor en alguna parte de la relación *cuenta* (cuyo nombre se ha cambiado a  $d$ ). El resultado contiene todos los saldos *salvo* el máximo. La Figura 2.16 muestra esta relación.

Paso 2: la consulta para determinar el saldo máximo de las cuentas del banco puede escribirse de la manera siguiente:

$$\Pi_{\text{saldo}}(\text{cuenta}) - \Pi_{\text{cuenta.saldo}} (\sigma_{\text{cuenta.saldo} < d.\text{saldo}} (\text{cuenta} \times \rho_d(\text{cuenta})))$$

<i>saldo</i>
350
400
500
700
750

**Figura 2.16** Resultado de la subexpresión

$$\Pi_{\text{cuenta.saldo}} (\sigma_{\text{cuenta.saldo} < d.\text{saldo}} (\text{cuenta} \times \rho_d(\text{cuenta}))).$$

saldo
900

**Figura 2.17** Saldo máximo de las cuentas del banco.

La Figura 2.17 muestra el resultado de esta consulta.

Considérese la consulta “Determinar el nombre de todos los clientes que viven en la misma ciudad y en la misma calle que Gómez” como un nuevo ejemplo de la operación renombramiento. Se puede obtener la calle y la ciudad en las que vive Gómez escribiendo

$$\Pi_{\text{calle\_cliente}, \text{ciudad\_cliente}} (\sigma_{\text{nombre\_cliente} = \text{“Gómez”}} (\text{cliente}))$$

Sin embargo, para hallar a otros clientes que vivan en esa ciudad y en esa calle hay que hacer referencia por segunda vez a la relación *cliente*. En la consulta siguiente se usa la operación renombramiento sobre la expresión anterior para darle al resultado el nombre *dirección\_gómez* y para cambiar el nombre de los atributos *calle\_cliente* y *ciudad\_cliente* a *calle* y *ciudad*, respectivamente:

$$\begin{aligned} & \Pi_{\text{cliente.nombre\_cliente}} \\ & (\sigma_{\text{cliente.calle\_cliente}=\text{dirección\_gómez.calle} \wedge \text{cliente.ciudad\_cliente}=\text{dirección\_gómez.ciudad}} \\ & (\text{cliente} \times \rho_{\text{dirección\_gómez(calle,ciudad)}} \\ & (\Pi_{\text{calle\_cliente}, \text{ciudad\_cliente}} (\sigma_{\text{nombre\_cliente} = \text{“Gómez”}} (\text{cliente})))))) \end{aligned}$$

El resultado de esta consulta, cuando se aplica a la relación *cliente* de la Figura 2.4, se muestra en la Figura 2.18.

La operación renombramiento no es estrictamente necesaria, dado que es posible usar una notación posicional para los atributos. Se pueden nombrar los atributos de una relación de manera implícita donde \$1, \$2, ... hagan referencia respectivamente al primer atributo, al segundo, etc. La notación posicional también se aplica a los resultados de las operaciones del álgebra relacional. La siguiente expresión del álgebra relacional ilustra el empleo de esta notación con el operador unario  $\sigma$ :

$$\sigma_{\$2=\$3}(R \times R)$$

Si una operación binaria necesita distinguir entre las dos relaciones que son sus operandos, se puede usar una notación posicional parecida para los nombres de las relaciones. Por ejemplo,  $\$R1$  puede hacer referencia al primer operando y  $\$R2$ , al segundo. Sin embargo, la notación posicional no resulta conveniente para las personas, dado que la posición del atributo es un número en vez de un nombre de atributo fácil de recordar. Por tanto, en este libro no se usa la notación posicional.

### 2.2.8 Definición formal del álgebra relacional

Las operaciones del Apartado 2.2 permiten dar una definición completa de las expresiones del álgebra relacional. Las expresiones fundamentales del álgebra relacional se componen de alguno de las siguientes elementos:

- Una relación de la base de datos
- Una relación constante

Las relaciones constantes se escriben poniendo una relación de sus tuplas entre llaves ( $\{ \}$ ), por ejemplo  $\{(C-101, \text{Centro}, 500) (C-215, \text{Becerril}, 700)\}$ .

nombre_cliente
Gómez
Pérez

**Figura 2.18** Clientes que viven en la misma ciudad y en la misma calle que Gómez.



<i>nombre_cliente</i>
Gómez
López
Santos

**Figura 2.19** Clientes con una cuenta abierta y un préstamo en el banco.

Las expresiones generales del álgebra relacional se construyen a partir de subexpresiones más pequeñas. Sean  $E_1$  y  $E_2$  expresiones de álgebra relacional. Todas las expresiones siguientes son también expresiones del álgebra relacional:

- $E_1 \cup E_2$
- $E_1 - E_2$
- $E_1 \times E_2$
- $\sigma_P(E_1)$ , donde  $P$  es un predicado de atributos de  $E_1$
- $\Pi_S(E_1)$ , donde  $S$  es una lista que se compone de algunos de los atributos de  $E_1$
- $\rho_x(E_1)$ , donde  $x$  es el nuevo nombre del resultado de  $E_1$

## 2.3 Otras operaciones del álgebra relacional

Las operaciones fundamentales del álgebra relacional son suficientes para expresar cualquier consulta del álgebra relacional<sup>1</sup>. Sin embargo, limitándose exclusivamente a las operaciones fundamentales, algunas consultas habituales resultan complicadas de expresar. Por tanto, se definen otras operaciones que no añaden potencia al álgebra, pero que simplifican las consultas habituales. Para cada operación nueva se facilita una expresión equivalente usando sólo las operaciones fundamentales.

### 2.3.1 Operación intersección de conjuntos

La primera operación adicional del álgebra relacional que se va a definir es la **intersección de conjuntos** ( $\cap$ ). Supóngase que se desea conocer todos los clientes con un préstamo concedido y una cuenta abierta. Empleando la intersección de conjuntos se puede escribir

$$\Pi_{\text{nombre\_cliente}}(\text{prestatarario}) \cap \Pi_{\text{nombre\_cliente}}(\text{impositor})$$

La relación resultante de esta consulta aparece en la Figura 2.19.

Obsérvese que se puede volver a escribir cualquier expresión del álgebra relacional que utilice 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  $r - (r - s)$ .

### 2.3.2 Operación reunión natural

Suele resultar deseable simplificar ciertas consultas que exijan un producto cartesiano. Generalmente, las consultas que implican un producto cartesiano incluyen un operador selección sobre el resultado del producto cartesiano. Considérese la consulta “Hallar los nombres de todos los clientes que tienen concedido un préstamo en el banco y averiguar su número e importe”. En primer lugar se calcula el

1. En el Apartado 2.4 se introducen operaciones que extienden la potencia del álgebra relacional al tratamiento de los valores nulos y de los valores de agregación.

<i>nombre_cliente</i>	<i>número_préstamo</i>	<i>importe</i>
Fernández	P-16	1.300
Gómez	P-23	2.000
Gómez	P-11	900
López	P-15	1.500
Pérez	P-93	500
Santos	P-17	1.000
Sotoca	P-14	1.500
Valdivieso	P-17	1.000

**Figura 2.20** Resultado de  $\Pi_{\text{nombre\_cliente}, \text{número\_préstamo}, \text{importe}} (\text{prestatario} \bowtie \text{préstamo})$ .

producto cartesiano de las relaciones *prestatario* y *préstamo*. Después se seleccionan las tuplas que sólo atañen al mismo *número\_préstamo*, seguidas por la proyección de *nombre\_cliente*, *número\_préstamo* e *importe* resultantes:

$$\Pi_{\text{nombre\_cliente}, \text{préstamo.número\_préstamo}, \text{importe}} (\sigma_{\text{prestatario.número\_préstamo} = \text{préstamo.número\_préstamo}} (\text{prestatario} \times \text{préstamo}))$$

La *reunión natural* es una operación binaria que permite combinar ciertas selecciones y un producto cartesiano en una sola operación. Se denota por el símbolo de **reunión**  $\bowtie$ . La operación reunión natural forma un producto cartesiano de sus dos argumentos, realiza una selección forzando la igualdad de los atributos que aparecen en ambos esquemas de relación y, finalmente, elimina los atributos duplicados.

Aunque la definición de la reunión natural es compleja, la operación es sencilla de aplicar. A modo de ilustración, considérese nuevamente el ejemplo “Determinar el nombre de todos los clientes que tienen concedido un préstamo en el banco y averiguar su importe”. Esta consulta puede expresarse usando la reunión natural de la manera siguiente:

$$\Pi_{\text{nombre\_cliente}, \text{número\_préstamo}, \text{importe}} (\text{prestatario} \bowtie \text{préstamo})$$

Dado que los esquemas de *prestatario* y de *préstamo* (es decir, *Esquema\_prestatario* y *Esquema\_préstamo*) tienen en común el atributo *número\_préstamo*, la operación reunión natural sólo considera los pares de tuplas que tienen el mismo valor de *número\_préstamo*. Esta operación combina cada uno de estos pares en una sola tupla en la unión de los dos esquemas (es decir, *nombre\_cliente*, *nombre\_sucursal*, *número\_préstamo*, *importe*). Después de realizar la proyección, se obtiene la relación mostrada en la Figura 2.20.

Considérense dos esquemas de relación *R* y *S*—que son, por supuesto, listas de nombres de atributos. Si se consideran los esquemas como *conjuntos*, en vez de como listas, se pueden denotar los nombres de los atributos que aparecen tanto en *R* como en *S* mediante  $R \cap S$ , y los nombres de los atributos que aparecen en *R*, en *S* o en ambos mediante  $R \cup S$ . De manera parecida, los nombres de los atributos que aparecen en *R* pero no en *S* se denotan por  $R - S$ , mientras que  $S - R$  denota los nombres de los atributos que aparecen en *S* pero no en *R*. Obsérvese que las operaciones unión, intersección y diferencia aquí operan sobre conjuntos de atributos, y no sobre relaciones.

Ahora es posible dar una definición formal de la reunión natural. Considérense dos relaciones  $r(R)$  y  $s(S)$ . La **reunión natural** de  $r$  y de  $s$ , denotada mediante  $r \bowtie s$  es una relación del esquema  $R \cup S$  definida formalmente de la manera siguiente:

$$r \bowtie s = \Pi_{R \cup S} (\sigma_{r.A_1 = s.A_1 \wedge r.A_2 = s.A_2 \wedge \dots \wedge r.A_n = s.A_n} (r \times s))$$

donde  $R \cap S = \{A_1, A_2, \dots, A_n\}$ .

Como la reunión natural es fundamental para gran parte de la teoría y de la práctica de las bases de datos relacionales, se ofrecen varios ejemplos de su uso.

- Hallar el nombre de todas las sucursales con clientes que tienen una cuenta abierta en el banco y viven en Peguerinos.

nombre_sucursal
Galapagar
Navacerrada

**Figura 2.21** Resultado de

$$\Pi_{\text{nombre\_sucursal}}(\sigma_{\text{ciudad\_cliente} = \text{"Peguerinos"}} (\text{cliente} \bowtie \text{cuenta} \bowtie \text{impositor})).$$

$$\Pi_{\text{nombre\_sucursal}}(\sigma_{\text{ciudad\_cliente} = \text{"Peguerinos"}} (\text{cliente} \bowtie \text{cuenta} \bowtie \text{impositor}))$$

La relación resultante de esta consulta aparece en la Figura 2.21.

Obsérvese que se ha escrito  $\text{cliente} \bowtie \text{cuenta} \bowtie \text{impositor}$  sin añadir paréntesis para especificar el orden en que se deben ejecutar las operaciones reunión natural sobre las tres relaciones. En el caso anterior hay dos posibilidades:

$$\begin{aligned} &(\text{cliente} \bowtie \text{cuenta}) \bowtie \text{impositor} \\ &\text{cliente} \bowtie (\text{cuenta} \bowtie \text{impositor}) \end{aligned}$$

No se ha especificado la expresión deseada, ya que las dos son equivalentes. Es decir, la reunión natural es **asociativa**.

- Hallar todos los clientes que tienen una cuenta abierta y un préstamo concedido en el banco.

$$\Pi_{\text{nombre\_cliente}} (\text{prestatario} \bowtie \text{impositor})$$

Obsérvese que en el Apartado 2.3.1 se escribió una expresión para esta consulta usando la intersección de conjuntos. A continuación se repite esa expresión.

$$\Pi_{\text{nombre\_cliente}} (\text{prestatario}) \cap \Pi_{\text{nombre\_cliente}} (\text{impositor})$$

La relación resultante de esta consulta se mostró anteriormente en la Figura 2.19. Este ejemplo ilustra una característica del álgebra relacional: se pueden escribir varias expresiones del álgebra relacional equivalentes que sean bastante diferentes entre sí.

- Sean  $r(R)$  y  $s(S)$  relaciones sin atributos en común; es decir,  $R \cap S = \emptyset$ . ( $\emptyset$  denota el conjunto vacío). Por tanto,  $r \bowtie s = r \times s$ .

La operación *reunión zeta* es una extensión de la operación reunión natural que permite combinar una selección y un producto cartesiano en una sola operación. Considérense las relaciones  $r(R)$  y  $s(S)$ , y sea  $\theta$  un predicado de los atributos del esquema  $R \cup S$ . La operación **reunión zeta**  $r \bowtie_{\theta} s$  se define de la manera siguiente:

$$r \bowtie_{\theta} s = \sigma_{\theta}(r \times s)$$

### 2.3.3 Operación división

La operación **división**, denotada por  $\div$ , resulta adecuada para las consultas que incluyen la expresión “para todos”. Supóngase que se desea hallar a todos los clientes que tengan abierta una cuenta en *todas* las sucursales ubicadas en Arganzuela. Se pueden obtener todas las sucursales de Arganzuela mediante la expresión

$$r_1 = \Pi_{\text{nombre\_sucursal}} (\sigma_{\text{ciudad\_sucursal} = \text{"Arganzuela"}} (\text{sucursal}))$$

La relación resultante de esta expresión aparece en la Figura 2.22.

nombre_sucursal
Centro
Galapagar

**Figura 2.22** Resultado de  $\Pi_{\text{nombre\_sucursal}}(\sigma_{\text{ciudad\_sucursal} = \text{"Arganzuela"}} (\text{sucursal}))$ .

Se pueden encontrar todos los pares (*nombre\_cliente*, *nombre\_sucursal*) para los que el cliente tiene una cuenta en una sucursal escribiendo

$$r_2 = \Pi_{\text{nombre\_cliente}, \text{nombre\_sucursal}} (\text{impositor} \bowtie \text{cuenta})$$

La Figura 2.23 muestra la relación resultante de esta expresión.

Ahora hay que hallar los clientes que aparecen en  $r_2$  con los nombres de *todas* las sucursales de  $r_1$ . La operación que proporciona exactamente esos clientes es la operación división. La consulta se formula escribiendo

$$\frac{\Pi_{\text{nombre\_cliente}, \text{nombre\_sucursal}} (\text{impositor} \bowtie \text{cuenta})}{\div \Pi_{\text{nombre\_sucursal}} (\sigma_{\text{ciudad\_sucursal} = \text{"Arganzuela"}} (\text{sucursal}))}$$

El resultado de esta expresión es una relación que tiene el esquema (*nombre\_cliente*) y que contiene la tupla (González).

Formalmente, sean  $r(R)$  y  $s(S)$  relaciones y  $S \subseteq R$ ; es decir, todos los atributos del esquema  $S$  están también en el esquema  $R$ . La relación  $r \div s$  es una relación del esquema  $R - S$  (es decir, del esquema que contiene todos los atributos del esquema  $R$  que no están en el esquema  $S$ ). Una tupla  $t$  está en  $r \div s$  si y sólo si se cumplen estas dos condiciones:

1.  $t$  está en  $\Pi_{R-S}(r)$
2. Para cada tupla  $t_s$  de  $s$  hay una tupla  $t_r$  de  $r$  que cumple las dos condiciones siguientes:
  - a.  $t_r[S] = t_s[S]$
  - b.  $t_r[R - S] = t$

Puede resultar sorprendente descubrir que, dados una operación división y los esquemas de las relaciones, se pueda definir la operación división en términos de las operaciones fundamentales. Sean  $r(R)$  y  $s(S)$  dadas, con  $S \subseteq R$ :

$$r \div s = \Pi_{R-S}(r) - \Pi_{R-S}((\Pi_{R-S}(r) \times s) - \Pi_{R-S,S}(r))$$

Para comprobar que esta expresión es cierta, obsérvese que  $\Pi_{R-S}(r)$  proporciona todas las tuplas  $t$  que cumplen la primera condición de la definición de la división. La expresión del lado derecho del operador diferencia de conjuntos,

$$\Pi_{R-S}((\Pi_{R-S}(r) \times s) - \Pi_{R-S,S}(r))$$

elimina las tuplas que no cumplen la segunda condición de la definición de la división. Esto se logra de la manera siguiente. Considérese  $\Pi_{R-S}(r) \times s$ . Esta relación está en el esquema  $R$  y empareja cada tupla de  $\Pi_{R-S}(r)$  con cada tupla de  $s$ . La expresión  $\Pi_{R-S,S}(r)$  sólo reordena los atributos de  $r$ .

Por tanto,  $(\Pi_{R-S}(r) \times s) - \Pi_{R-S,S}(r)$  genera los pares de tuplas de  $\Pi_{R-S}(r)$  y de  $s$  que no aparecen en  $r$ . Si una tupla  $t_j$  está en

$$\Pi_{R-S}((\Pi_{R-S}(r) \times s) - \Pi_{R-S,S}(r))$$

<i>nombre_cliente</i>	<i>nombre_sucursal</i>
Abril	Collado Mediano
Gómez	Becerril
González	Centro
González	Galapagar
López	Navacerrada
Rupérez	Moralzarzal
Santos	Galapagar

**Figura 2.23** Resultado de  $\Pi_{\text{nombre\_cliente}, \text{nombre\_sucursal}} (\text{impositor} \bowtie \text{cuenta})$ .

<i>nombre_cliente</i>	<i>límite</i>	<i>saldo_crédito</i>
Gómez	2.000	400
López	1.500	1.500
Pérez	2.000	1.750
Santos	6.000	700

**Figura 2.24** La relación *información\_crédito*.

existe alguna tupla  $t_s$  de  $s$  que no se combina con la tupla  $t_j$  para formar una tupla de  $r$ . Por tanto,  $t_j$  guarda un valor de los atributos  $R - S$  que no aparece en  $r \div s$ . Estos valores son los que se eliminan de  $\Pi_{R-S}(r)$ .

### 2.3.4 Operación asignación

En ocasiones resulta conveniente escribir una expresión del álgebra relacional mediante la asignación de partes de esa expresión a variables de relación temporal. La operación **asignación**, denotada por  $\leftarrow$ , actúa de manera parecida a la asignación de los lenguajes de programación. Para ilustrar esta operación, considérese la definición de la división dada en el Apartado 2.3.3. Se puede escribir  $r \div s$  como

$$\begin{aligned} temp1 &\leftarrow \Pi_{R-S}(r) \\ temp2 &\leftarrow \Pi_{R-S}((temp1 \times s) - \Pi_{R-S,S}(r)) \\ resultado &= temp1 - temp2 \end{aligned}$$

La evaluación de una asignación no hace que se muestre ninguna relación al usuario. Por el contrario, el resultado de la expresión situada a la derecha de  $\leftarrow$  se asigna a la variable relación situada a la izquierda de  $\leftarrow$ . Esta variable relación puede usarse en expresiones posteriores.

Con la operación asignación se pueden escribir las consultas como programas secuenciales que consisten de una serie de asignaciones seguida de una expresión cuyo valor se muestra como resultado de la consulta. En las consultas del álgebra relacional la asignación siempre debe hacerse a una variable de relación temporal. Las asignaciones a relaciones permanentes constituyen una modificación de la base de datos. Este asunto se estudiará en el Apartado 2.6. Obsérvese que la operación asignación no añade potencia alguna al álgebra. Resulta, sin embargo, una manera conveniente de expresar las consultas complejas.

## 2.4 Operaciones del álgebra relacional extendida

Las operaciones básicas del álgebra relacional se han extendido de varias formas. Una sencilla es permitir operaciones aritméticas como parte de las proyecciones. Una importante es permitir *operaciones de agregación*, como el cálculo de la suma de los elementos de un conjunto, o su media. Otra extensión importante es la operación *reunión externa*, que permite a las expresiones del álgebra relacional trabajar con los valores nulos, que modelan la información ausente.

### 2.4.1 Proyección generalizada

La operación **proyección generalizada** extiende la proyección permitiendo que se utilicen funciones aritméticas en la lista de proyección. La operación proyección generalizada es de la forma

$$\Pi_{F_1, F_2, \dots, F_n}(E)$$

donde  $E$  es cualquier expresión del álgebra relacional y  $F_1, F_2, \dots, F_n$  son expresiones aritméticas que incluyen constantes y atributos del esquema de  $E$ . Como caso especial, la expresión aritmética puede ser simplemente un atributo o una constante.

Por ejemplo, supóngase que se dispone de una relación *información\_crédito*, como se muestra en la Figura 2.24, que proporciona el límite de crédito y el importe consumido actualmente (*saldo\_crédito*). Si se desea determinar el importe disponible por cada persona, se puede escribir la expresión siguiente:

<i>nombre_cliente</i>	<i>crédito_disponible</i>
Gómez	1.600
López	0
Pérez	250
Santos	5.300

**Figura 2.25** Resultado de  $\Pi_{\text{nombre\_cliente}, (\text{límite} - \text{saldo\_crédito}) \text{ as } \text{crédito\_disponible}} (\text{información\_crédito})$ .

$$\Pi_{\text{nombre\_cliente}, \text{límite} - \text{saldo\_crédito}} (\text{información\_crédito})$$

El atributo resultante de la expresión  $\text{límite} - \text{saldo\_crédito}$  no tiene nombre asignado. Se puede aplicar la operación renombramiento al resultado de la proyección generalizada para darle nombre. Como conveniencia notacional, el renombramiento de atributos se puede combinar con la proyección generalizada como se ilustra a continuación:

$$\Pi_{\text{nombre\_cliente}, (\text{límite} - \text{saldo\_crédito}) \text{ as } \text{crédito\_disponible}} (\text{información\_crédito})$$

Al segundo atributo de esta proyección generalizada se le ha asignado el nombre *crédito\_disponible*. La Figura 2.25 muestra el resultado de aplicar esta expresión a la relación de la Figura 2.24.

## 2.4.2 Funciones de agregación

Las **funciones de agregación** toman un conjunto de valores y devuelven como resultado un único valor. Por ejemplo, la función de agregación **sum** toma un conjunto de valores y devuelve su suma. Por tanto, la función **sum** aplicada al conjunto

$$\{1, 1, 3, 4, 4, 11\}$$

devuelve el valor 24. La función de agregación **avg** devuelve la media de los valores. Cuando se aplica al conjunto anterior devuelve el valor 4. La función de agregación **count** devuelve el número de elementos del conjunto, y devolvería 6 en el caso anterior. Otras funciones de agregación habituales son **min** y **max**, que devuelven los valores mínimo y máximo del conjunto; en el ejemplo anterior devuelven 1 y 11, respectivamente.

Los conjuntos sobre los que operan las funciones de agregación pueden contener valores repetidos; el orden en el que aparezcan los valores no tiene importancia. Estos conjuntos se denominan **multiconjuntos**. Los conjuntos son un caso especial de los multiconjuntos, en los que sólo hay una copia de cada elemento.

Para ilustrar el concepto de agregación se usará la relación *trabajo\_por\_horas* descrita en la Figura 2.26, que muestra los empleados a tiempo parcial. Supóngase que se desea determinar la suma total de los sueldos de los empleados del banco a tiempo parcial. La expresión del álgebra relacional para esta consulta es la siguiente:

$$\mathcal{G}_{\text{sum}(\text{sueldo})}(\text{trabajo\_por\_horas})$$

<i>nombre_empleado</i>	<i>nombre_sucursal</i>	<i>sueldo</i>
Cana	Leganés	1.500
Cascallar	Navacerrada	5.300
Catalán	Leganés	1.600
Díaz	Centro	1.300
Fernández	Navacerrada	1.500
González	Centro	1.500
Jiménez	Centro	2.500
Ribera	Navacerrada	1.300

**Figura 2.26** La relación *trabajo\_por\_horas*.

<i>nombre_empleado</i>	<i>nombre_sucursal</i>	<i>sueldo</i>
Cana	Leganés	1.500
Catalán	Leganés	1.600
Díaz	Centro	1.300
González	Centro	1.500
Jiménez	Centro	2.500
Cascallar	Navacerrada	5.300
Fernández	Navacerrada	1.500
Ribera	Navacerrada	1.300

**Figura 2.27** La relación *trabajo\_por\_horas* después de la agrupación.

El símbolo  $\mathcal{G}$  es la letra G en el tipo de letra caligráfico; se lee “G caligráfica”. La operación del álgebra relacional  $\mathcal{G}$  significa que se debe aplicar la agrupación; y su subíndice, la operación de agrupación que se aplica. El resultado de la expresión anterior es una relación con un único atributo, que contiene una sola fila con un valor numérico correspondiente a la suma de los sueldos de todos los trabajadores que trabajan en el banco a tiempo parcial.

Supóngase ahora que se desea determinar la suma total de sueldos de los empleados a tiempo parcial de cada sucursal bancaria, en lugar de determinar la suma total. Para ello hay que dividir la relación *trabajo\_por\_horas* en **grupos** basados en la sucursal y aplicar la función de agrupación a cada grupo. La expresión siguiente obtiene el resultado deseado usando el operador de agrupación  $\mathcal{G}$ :

$$\text{nombre\_sucursal} \mathcal{G}_{\text{sum}(\text{sueldo})}(\text{trabajo\_por\_horas})$$

Existen casos en los que es necesario borrar los valores repetidos antes de calcular una función de agrupación. Si se desea eliminar los valores repetidos hay que usar los mismos nombres de funciones que antes, con la cadena de texto “**distinct**” precedida de un guión añadida al final del nombre de la función (por ejemplo, **count-distinct**). Un ejemplo se da en la consulta “Determinar el número de sucursales que aparecen en la relación *trabajo\_por\_horas*”. En este caso, el nombre de cada sucursal sólo se cuenta una vez, independientemente del número de empleados que trabajen en la misma. Esta consulta se escribe de la manera siguiente:

$$\mathcal{G}_{\text{count-distinct}(\text{nombre\_sucursal})}(\text{trabajo\_por\_horas})$$

Al aplicar esta consulta a la relación de la Figura 2.26 el resultado está compuesto por una única fila con valor 3. En esta consulta el atributo *nombre\_sucursal* en el subíndice a la izquierda de  $\mathcal{G}$  indica que la relación de entrada *trabajo\_por\_horas* debe dividirse en grupos de acuerdo con el valor de *nombre\_sucursal*. La Figura 2.27 muestra los grupos resultantes. La expresión  $\text{sum}(\text{sueldo})$  en el subíndice a la derecha de  $\mathcal{G}$  indica que, para cada grupo de tuplas (es decir, para cada sucursal) hay que aplicar la función de agrupación **sum** al conjunto de valores del atributo *sueldo*. La relación resultante consiste en tuplas con el nombre de la sucursal y la suma de los sueldos de la sucursal, como se muestra en la Figura 2.28.

La forma general de la **operación de agrupación**  $\mathcal{G}$  es la siguiente:

$$G_1, G_2, \dots, G_n \mathcal{G}_{F_1(A_1), F_2(A_2), \dots, F_m(A_m)}(E)$$

<i>nombre_sucursal</i>	<i>sum(sueldo)</i>
Centro	5.300
Leganés	3.100
Navacerrada	8.100

**Figura 2.28** Resultado de  $\text{nombre\_sucursal} \mathcal{G}_{\text{sum}(\text{sueldo})}(\text{trabajo\_por\_horas})$ .



<i>nombre_sucursal</i>	<i>suma_sueldo</i>	<i>sueldo_máximo</i>
Centro	5.300	2.500
Leganés	3.100	1.600
Navacerrada	8.100	5.300

**Figura 2.29** Resultado de  $\text{nombre\_sucursal } \mathcal{G}_{\text{sum}(\text{sueldo})} \text{ as suma\_sueldo, max}(\text{sueldo}) \text{ as sueldo\_máximo}(\text{trabajo\_por\_horas})$ .

donde  $E$  es cualquier expresión del álgebra relacional;  $G_1, G_2, \dots, G_n$  constituye una lista de atributos sobre los que se realiza la agrupación, cada  $F_i$  es una función de agregación y cada  $A_i$  es el nombre de un atributo. El significado de la operación es el siguiente: las tuplas del resultado de la expresión  $E$  se dividen en grupos de manera que

1. Todas las tuplas de cada grupo tienen los mismos valores para  $G_1, G_2, \dots, G_n$ .
2. Las tuplas de grupos diferentes tienen valores diferentes para  $G_1, G_2, \dots, G_n$ .

Por tanto, los grupos pueden identificarse por el valor de los atributos  $G_1, G_2, \dots, G_n$ . Para cada grupo  $(g_1, g_2, \dots, g_n)$  el resultado tiene una tupla  $(g_1, g_2, \dots, g_n, a_1, a_2, \dots, a_m)$  donde, para cada  $i$ ,  $a_i$  es el resultado de aplicar la función de agregación  $F_i$  al multiconjunto de valores del atributo  $A_i$  en el grupo.

Como caso especial de la operación de agregación, la lista de atributos  $G_1, G_2, \dots, G_n$  puede estar vacía, en cuyo caso sólo hay un grupo que contiene todas las tuplas de la relación. Esto se corresponde con la agregación sin agrupación.

Volviendo al ejemplo anterior, si se deseara determinar el sueldo máximo de los empleados a tiempo parcial de cada sucursal, además de la suma de los sueldos, habría que escribir la expresión

$$\text{nombre\_sucursal } \mathcal{G}_{\text{sum}(\text{sueldo}), \text{max}(\text{sueldo})}(\text{trabajo\_por\_horas})$$

Como en la proyección generalizada, el resultado de las operaciones de agregación no tiene nombre. Se puede aplicar la operación renombramiento al resultado para darle un nombre. Como conveniencia notacional, los atributos de las operaciones de agregación se pueden renombrar como se indica a continuación:

$$\text{nombre\_sucursal } \mathcal{G}_{\text{sum}(\text{sueldo})} \text{ as suma\_sueldo, max}(\text{sueldo}) \text{ as sueldo\_máximo}(\text{trabajo\_por\_horas})$$

La Figura 2.29 muestra el resultado de la expresión.

### 2.4.3 Reunión externa

La operación **reunión externa** es una ampliación de la operación reunión para trabajar con información ausente. Supóngase que se dispone de relaciones con los siguientes esquemas y que contienen datos de empleados a tiempo completo:

$$\begin{aligned} &\text{empleado}(\text{nombre\_empleado}, \text{calle}, \text{ciudad}) \\ &\text{trabajo\_a\_tiempo\_completo}(\text{nombre\_empleado}, \text{nombre\_sucursal}, \text{sueldo}) \end{aligned}$$

Considérense las relaciones *empleado* y *trabajo\_a\_tiempo\_completo* mostradas en la Figura 2.30. Supóngase que se desea generar una única relación con toda la información (calle, ciudad, nombre de la sucursal y sueldo) de los empleados a tiempo completo. Un posible enfoque sería usar la operación reunión natural de la manera siguiente:

$$\text{empleado} \bowtie \text{trabajo\_a\_tiempo\_completo}$$

El resultado de esta expresión se muestra en la Figura 2.31. Obsérvese que se ha perdido la información sobre la calle y la ciudad de residencia de Gómez, dado que la tupla que describe a Gómez no está presente en la relación *trabajo\_a\_tiempo\_completo*; de manera parecida, se ha perdido la información sobre el nombre de la sucursal y sobre el sueldo de Barea, dado que la tupla que describe a Barea no está presente en la relación *empleado*.

<i>nombre_empleado</i>	<i>calle</i>	<i>ciudad</i>
Segura	Tebeo	La Loma
Domínguez	Viaducto	Villaconejos
Gómez	Bailén	Alcorcón
Valdivieso	Fuencarral	Móstoles

<i>nombre_empleado</i>	<i>nombre_sucursal</i>	<i>sueldo</i>
Segura	Majadahonda	1.500
Domínguez	Majadahonda	1.300
Barea	Fuenlabrada	5.300
Valdivieso	Fuenlabrada	1.500

**Figura 2.30** Las relaciones *empleado* y *trabajo\_a\_tiempo\_completo*.

<i>nombre_empleado</i>	<i>calle</i>	<i>ciudad</i>	<i>nombre_sucursal</i>	<i>sueldo</i>
Domínguez	Viaducto	Villaconejos	Majadahonda	1.300
Segura	Tebeo	La Loma	Majadahonda	1.500
Valdivieso	Fuencarral	Móstoles	Fuenlabrada	1.500

**Figura 2.31** El resultado de *empleado*  $\bowtie$  *trabajo\_a\_tiempo\_completo*.

Se puede usar la operación reunión externa para evitar esta pérdida de información. En realidad, esta operación tiene tres formas diferentes: *reunión externa por la izquierda*, denotada por  $\bowtie\!\!\!\bowtie$ ; *reunión externa por la derecha*, denotada por  $\bowtie\!\!\!\bowtie$  y *reunión externa completa*, denotada por  $\bowtie\!\!\!\bowtie$ . Las tres formas de la reunión externa calculan la reunión y añaden tuplas adicionales al resultado de la misma. El resultado de las expresiones *empleado*  $\bowtie\!\!\!\bowtie$  *trabajo\_a\_tiempo\_completo*, *empleado*  $\bowtie\!\!\!\bowtie$  *trabajo\_a\_tiempo\_completo* y *empleado*  $\bowtie\!\!\!\bowtie$  *trabajo\_a\_tiempo\_completo* se muestra en las Figuras 2.32, 2.33 y 2.34, respectivamente.

La **reunión externa por la izquierda** ( $\bowtie\!\!\!\bowtie$ ) toma todas las tuplas de la relación de la izquierda que no coinciden con ninguna tupla de la relación de la derecha, las rellena con valores nulos en todos los demás atributos de la relación de la derecha y las añade al resultado de la reunión natural. En la Figura 2.32 la tupla (Gómez, Bailén, Alcorcón, *nulo*, *nulo*) es una tupla de este tipo. Toda la información de la relación de la izquierda se halla presente en el resultado de la reunión externa por la izquierda.

La **reunión externa por la derecha** ( $\bowtie\!\!\!\bowtie$ ) es simétrica de la reunión externa por la izquierda. Rellena con valores nulos las tuplas de la relación de la derecha que no coinciden con ninguna tupla de la relación de la izquierda y las añade al resultado de la reunión natural. En la Figura 2.33 la tupla (Barea, *nulo*,

<i>nombre_empleado</i>	<i>calle</i>	<i>ciudad</i>	<i>nombre_sucursal</i>	<i>sueldo</i>
Domínguez	Viaducto	Villaconejos	Majadahonda	1.300
Gómez	Bailén	Alcorcón	<i>nulo</i>	<i>nulo</i>
Segura	Tebeo	La Loma	Majadahonda	1.500
Valdivieso	Fuencarral	Móstoles	Fuenlabrada	1.500

**Figura 2.32** Resultado de *empleado*  $\bowtie\!\!\!\bowtie$  *trabajo\_a\_tiempo\_completo*.

<i>nombre_empleado</i>	<i>calle</i>	<i>ciudad</i>	<i>nombre_sucursal</i>	<i>sueldo</i>
Barea	<i>nulo</i>	<i>nulo</i>	Fuenlabrada	5.300
Domínguez	Viaducto	Villaconejos	Majadahonda	1.300
Segura	Tebeo	La Loma	Majadahonda	1.500
Valdivieso	Fuencarral	Móstoles	Fuenlabrada	1.500

**Figura 2.33** Resultado de *empleado*  $\bowtie\!\!\!\bowtie$  *trabajo\_a\_tiempo\_completo*.

<i>nombre_empleado</i>	<i>calle</i>	<i>ciudad</i>	<i>nombre_sucursal</i>	<i>sueldo</i>
Barea	<i>nulo</i>	<i>nulo</i>	Fuenlabrada	5.300
Domínguez	Viaducto	Villaconejos	Majadahonda	1.300
Gómez	Bailén	Alcorcón	<i>nulo</i>	<i>nulo</i>
Segura	Tebeo	La Loma	Majadahonda	1.500
Valdivieso	Fuencarral	Móstoles	Fuenlabrada	1.500

**Figura 2.34** Resultado de  $\text{empleado} \bowtie \text{trabajo\_a\_tiempo\_parcial}$ .

*nulo*, Fuenlabrada, 5.300) es una tupla de este tipo. Por tanto, toda la información de la relación de la derecha se halla presente en el resultado de la reunión externa por la derecha.

La **reunión externa completa** ( $\bowtie$ ) realiza estas dos operaciones, rellenando las tuplas de la relación de la izquierda que no coinciden con ninguna tupla de la relación de la derecha y las tuplas de la relación de la derecha que no coinciden con ninguna tupla de la relación de la izquierda, y añadiéndolas al resultado de la reunión. La Figura 2.34 muestra el resultado de una reunión externa completa.

Puesto que las operaciones de reunión pueden generar resultados que contengan valores nulos, es necesario especificar la manera en que deben manejar estos valores las diferentes operaciones del álgebra relacional. El Apartado 2.5 aborda este problema.

Es interesante observar que las operaciones de reunión externa pueden expresarse mediante las operaciones básicas del álgebra relacional. Por ejemplo, la operación de reunión externa por la izquierda  $r \bowtie s$  se puede expresar como:

$$(r \bowtie s) \cup (r - \Pi_R(r \bowtie s)) \times \{(nulo, \dots, nulo)\}$$

donde la relación constante  $\{(nulo, \dots, nulo)\}$  se encuentra en el esquema  $S - R$ .

## 2.5 Valores nulos

En este apartado se define la forma en que las diferentes operaciones del álgebra relacional tratan los valores nulos y las complicaciones que surgen cuando los valores nulos participan en las operaciones aritméticas o en las comparaciones. Como se verá, a menudo hay varias formas de tratar los valores nulos y, en consecuencia, las siguientes definiciones pueden ser a veces arbitrarias. Las operaciones y las comparaciones con valores nulos se deben evitar siempre que sea posible.

Dado que el valor especial *nulo* indica “valor desconocido o no existente”, cualquier operación aritmética (como  $+$ ,  $-$ ,  $*$  y  $/$ ) que incluya valores nulos debe devolver un valor nulo.

De manera parecida, cualquier comparación (como  $<$ ,  $<=$ ,  $>$ ,  $>=$  y  $\neq$ ) que incluya un valor nulo tiene como resultado el valor especial **desconocido**; no se puede decir si el resultado de la comparación es cierto o falso, así que se dice que el resultado es el nuevo valor lógico *desconocido*.

Las comparaciones que incluyen valores nulos pueden aparecer dentro de expresiones booleanas que incluyan las operaciones y (conjunción), o (disyunción) y no (negación). Por tanto, se debe definir la forma en que estas tres operaciones lógicas tratan el valor lógico *desconocido*.

- **Y:** (*cierto y desconocido*) = *desconocido*; (*falso y desconocido*) = *falso*; (*desconocido y desconocido*) = *desconocido*.
- **O:** (*cierto o desconocido*) = *cierto*; (*falso o desconocido*) = *desconocido*; (*desconocido o desconocido*) = *desconocido*.
- **No:** (**no** *desconocido*) = *desconocido*.

Ahora es posible describir la forma en que las diferentes operaciones del álgebra relacional tratan los valores nulos.

- **Selección.** La operación selección evalúa el predicado  $P$  en  $\sigma_P(E)$  sobre cada tupla  $t$  de  $E$ . Si el predicado devuelve el valor *cierto*, se añade  $t$  al resultado. En caso contrario, si el predicado devuelve *desconocido* o *falso*,  $t$  no se añade al resultado.

- **Reunión.** Las reuniones se pueden expresar como un producto cartesiano seguido de una selección. Por tanto, la definición de la forma en que la selección trata los nulos también define la forma en que lo hacen las operaciones reunión.

En una reunión natural, por ejemplo,  $r \bowtie s$  se puede observar de la definición anterior que si dos tuplas,  $t_r \in r$  y  $t_s \in s$ , tienen un valor nulo en un atributo común, entonces las tuplas no coinciden.

- **Proyección.** La operación proyección trata los valores nulos como cualesquiera otros valores al eliminar valores duplicados. Así, si dos tuplas del resultado de la proyección son exactamente iguales y las dos tienen valores nulos en los mismos campos, se tratan como duplicados.

Esta decisión es un tanto arbitraria ya que, sin conocer el valor real, no se sabe si los dos valores nulos son duplicados o no.

- **Unión, intersección y diferencia.** Estas operaciones tratan los valores nulos igual que la operación proyección; tratan las tuplas que tienen los mismos valores en todos los campos como duplicados, aunque algunos de los campos tengan valores nulos en ambas tuplas.

Este comportamiento es un tanto arbitrario, especialmente en el caso de la intersección y la diferencia, dado que no se sabe si los valores reales (si existen) representados por los nulos son iguales.

- **Proyección generalizada.** Se describió la manera en que se tratan los nulos en las expresiones al comienzo del Apartado 2.5. Las tuplas duplicadas que contienen valores nulos se tratan como en la operación proyección.

- **Funciones de agregación.** Cuando hay valores nulos en los atributos agregados, las operaciones de agregación los tratan igual que en el caso de la proyección: si dos tuplas son iguales en todos los atributos de agregación, la operación las coloca en el mismo grupo, aunque parte de los valores de los atributos sean valores nulos.

Cuando aparecen valores nulos en los atributos agregados, la operación borra los valores nulos del resultado antes de aplicar la agregación. Si el multiconjunto resultante está vacío, el resultado agregado es nulo.

Obsérvese que el tratamiento de los valores nulos en este caso es diferente que en las expresiones aritméticas ordinarias; se podría haber definido el resultado de una operación de agregación como nulo si tan sólo uno de los valores agregados fuera nulo. Sin embargo, esto significaría que un único valor desconocido en un gran grupo podría hacer que el resultado agregado sobre el grupo fuese nulo, y se perdería una gran cantidad de información útil.

- **Reunión externa.** Las operaciones de reunión externa se comportan igual que las operaciones de reunión, excepto sobre las tuplas que no aparecen en el resultado de la reunión. Esas tuplas se pueden añadir al resultado (dependiendo de si la operación es  $\bowtie$ ,  $\bowtie_{\text{L}}$  o  $\bowtie_{\text{R}}$ ) rellenando con valores nulos.

## 2.6 Modificación de la base de datos

Hasta ahora se ha centrado la atención en la extracción de información de la base de datos. En este apartado se abordará la manera de añadir, eliminar y modificar información de la base de datos.

Las modificaciones de la base de datos se expresan mediante la operación asignación. Las asignaciones a las relaciones reales de la base de datos se realizan empleando la misma notación que se describió para la asignación en el Apartado 2.3.

### 2.6.1 Borrado

Las solicitudes de borrado se expresan básicamente igual que las consultas. Sin embargo, en lugar de mostrar las tuplas al usuario, se eliminan de la base de datos las tuplas seleccionadas. Sólo se pueden borrar tuplas enteras; no se pueden borrar valores de atributos concretos. En el álgebra relacional los borrados se expresan mediante

$$r \leftarrow r - E$$

donde  $r$  es una relación y  $E$  es una consulta del álgebra relacional.

A continuación se muestran varios ejemplos de borrado:

- Borrar todas las cuentas de Gómez.

$$\text{impositor} \leftarrow \text{impositor} - \sigma_{\text{nombre\_cliente} = \text{"Gómez"}}(\text{impositor})$$

- Borrar todos los préstamos con importes entre 0 y 50.

$$\text{préstamo} \leftarrow \text{préstamo} - \sigma_{\text{importe} \geq 0 \wedge \text{importe} \leq 50}(\text{préstamo})$$

- Borrar todas las cuentas de las sucursales de Arganzuela

$$\begin{aligned} r_1 &\leftarrow \sigma_{\text{ciudad\_sucursal} = \text{"Arganzuela"}}(\text{cuenta} \bowtie \text{sucursal}) \\ r_2 &\leftarrow \Pi_{\text{nombre\_sucursal}, \text{número\_cuenta}, \text{saldo}}(r_1) \\ \text{cuenta} &\leftarrow \text{cuenta} - r_2 \end{aligned}$$

Obsérvese que en el último ejemplo se ha simplificado la expresión mediante la asignación a relaciones temporales ( $r_1$  y  $r_2$ ).

## 2.6.2 Inserción

Para insertar datos en una relación hay que especificar la tupla que se va a insertar o escribir una consulta cuyo resultado sea el conjunto de tuplas que se van a insertar. Evidentemente, el valor de los atributos de las tuplas insertadas debe ser miembro del dominio de cada atributo. De manera parecida, las tuplas insertadas deben tener la aridad correcta. El álgebra relacional expresa las inserciones mediante

$$r \leftarrow r \cup E$$

donde  $r$  es una relación y  $E$  es una expresión del álgebra relacional. La inserción de una sola tupla se expresa haciendo que  $E$  sea una relación constante que contiene una tupla.

Supóngase que se desea insertar el hecho de que Gómez tiene 1.200 € en la cuenta C-973 de la sucursal de Navacerrada. Hay que escribir

$$\begin{aligned} \text{cuenta} &\leftarrow \text{cuenta} \cup \{(C-973, \text{"Navacerrada"}, 1200)\} \\ \text{impositor} &\leftarrow \text{impositor} \cup \{(\text{"Gómez"}, C-973)\} \end{aligned}$$

De forma más general, puede que se desee insertar tuplas según el resultado de una consulta. Supóngase que se desea ofrecer una nueva cuenta de ahorro con 200 € como regalo a todos los clientes con préstamos concedidos en la sucursal de Navacerrada. Se usará el número de préstamo como número de esta cuenta de ahorro. Se escribe:

$$\begin{aligned} r_1 &\leftarrow (\sigma_{\text{nombre\_sucursal} = \text{"Navacerrada"}}(\text{prestario} \bowtie \text{préstamo})) \\ r_2 &\leftarrow \Pi_{\text{número\_préstamo}, \text{nombre\_sucursal}}(r_1) \\ \text{cuenta} &\leftarrow \text{cuenta} \cup (r_2 \times \{(200)\}) \\ \text{impositor} &\leftarrow \text{impositor} \cup \Pi_{\text{nombre\_cliente}, \text{número\_préstamo}}(r_1) \end{aligned}$$

En lugar de especificar una tupla como se hizo anteriormente, se especifica un conjunto de tuplas que se inserta en las relaciones *cuenta* e *impositor*. Cada tupla de la relación *cuenta* tiene un *número\_cuenta* (que es igual que el número de préstamo), un *nombre\_sucursal* (Navacerrada) y el saldo inicial de la nueva cuenta (200 €). Cada tupla de la relación *impositor* tiene como *nombre\_cliente* el nombre del prestatario al que se le regala la nueva cuenta y el mismo número de cuenta que la correspondiente tupla de *cuenta*.

## 2.6.3 Actualización

Puede que, en algunas situaciones, se desee modificar un valor de una tupla sin modificar *todos* los valores de esa tupla. Se puede usar el operador proyección generalizada para llevar a cabo esta tarea:

$$r \leftarrow \Pi_{F_1, F_2, \dots, F_n}(r)$$

donde cada  $F_i$  es, o bien el  $i$ -ésimo atributo de  $r$  en el caso de que este atributo no se vaya a actualizar o, en caso contrario, una expresión sólo con constantes y atributos de  $r$  que proporciona el nuevo valor del atributo. Téngase en cuenta que el esquema de la expresión resultante de la expresión de proyección generalizada debe coincidir con el esquema original de  $r$ .

Si se desea seleccionar varias tuplas de  $r$  y actualizar sólo esas tuplas, se puede usar la expresión siguiente, donde  $P$  denota la condición de selección que escoge las tuplas que hay que actualizar:

$$r \leftarrow \Pi_{F_1, F_2, \dots, F_n}(\sigma_P(r)) \cup (r - \sigma_P(r))$$

Para ilustrar el uso de la operación actualización supóngase que se va a realizar el pago de los intereses y que hay que aumentar todos los saldos en un cinco por ciento:

$$cuenta \leftarrow \Pi_{\text{número\_cuenta}, \text{nombre\_sucursal}, \text{saldo} * 1.05}(cuenta)$$

Supóngase ahora que las cuentas con saldos superiores a 10.000 € reciben un interés del seis por ciento, mientras que el resto recibe un cinco por ciento:

$$\begin{aligned} cuenta \leftarrow & \Pi_{\text{número\_cuenta}, \text{nombre\_sucursal}, \text{saldo} * 1.06}(\sigma_{\text{saldo} > 10000}(cuenta)) \\ & \cup \Pi_{\text{número\_cuenta}, \text{nombre\_sucursal}, \text{saldo} * 1.05}(\sigma_{\text{saldo} \leq 10000}(cuenta)) \end{aligned}$$

## 2.7 Resumen

- El **modelo de datos relacional** se basa en un conjunto de tablas. El usuario del sistema de bases de datos puede consultar esas tablas, insertar tuplas nuevas, borrar tuplas y actualizar (modificar) las tuplas. Hay varios lenguajes para expresar estas operaciones.
- El **álgebra relacional** define un conjunto de operaciones algebraicas que operan sobre las tablas y devuelven tablas como resultado. Estas operaciones se pueden combinar para obtener expresiones que definan las consultas deseadas. El álgebra define las operaciones básicas usadas en los lenguajes de consultas relacionales.
- Las operaciones del álgebra relacional se pueden dividir en:
  - ☐ Operaciones básicas.
  - ☐ Operaciones adicionales que se pueden expresar en términos de las operaciones básicas.
  - ☐ Operaciones extendidas, algunas de las cuales añaden mayor poder expresivo al álgebra relacional.
- Las bases de datos se pueden modificar con la **inserción**, el **borrado** y la **actualización** de tuplas. Se ha usado el álgebra relacional con el **operador asignación** para expresar estas modificaciones.
- El álgebra relacional es un lenguaje rígido y formal que no resulta adecuado para los usuarios ocasionales de los sistemas de bases de datos. Los sistemas comerciales de bases de datos, por tanto, usan lenguajes con más “azúcar sintáctico”. En los Capítulos 3 y 4 se tomará en consideración el lenguaje más influyente, **SQL**, que está basado en el álgebra relacional.

## Términos de repaso

- Tabla.
- Relación.
- Variable tupla.
- Dominio atómico.
- Valor nulo.
- Esquema de la base de datos.
- Ejemplar de la base de datos.
- Esquema de la relación.
- Ejemplar de la relación.
- Claves.
- Clave externa.
  - ☐ Relación referenciante.
  - ☐ Relación referenciada.
- Diagrama de esquema.
- Lenguaje de consultas.
- Lenguaje procedimental.

- Lenguaje no procedimental.
- Álgebra relacional.
- Operaciones del álgebra relacional:
  - ☐ Selección ( $\sigma$ ).
  - ☐ Proyección ( $\Pi$ ).
  - ☐ Unión ( $\cup$ ).
  - ☐ Diferencia de conjuntos ( $-$ ).
  - ☐ Producto cartesiano ( $\times$ ).
  - ☐ Renombramiento ( $\rho$ ).
- Operaciones adicionales:
  - ☐ Intersección de conjuntos ( $\cap$ ).
  - ☐ Reunión natural ( $\bowtie$ ).
  - ☐ División ( $/$ ).
- Operación asignación.
- Operaciones del álgebra relacional extendida:
  - ☐ Proyección generalizada ( $\Pi$ ).
  - ☐ Reunión externa.
    - Reunión externa por la izquierda ( $\bowtie$ ).
    - Reunión externa por la derecha ( $\bowtie$ ).
    - Reunión externa completa ( $\bowtie$ ).
  - ☐ Agregación ( $\mathcal{G}$ ).
- Multiconjuntos.
- Agrupación.
- Valores nulos.
- Valores lógicos:
  - ☐ *cierto*.
  - ☐ *falso*.
  - ☐ *desconocido*.
- Modificación de la base de datos.
  - ☐ Borrado.
  - ☐ Inserción.
  - ☐ Actualización.

## Ejercicios prácticos

- 2.1 Considérese la base de datos relacional de la Figura 2.35, en la que las claves primarias están subrayadas. Obténgase una expresión del álgebra relacional para cada una de las consultas siguientes:
- a. Determinar el nombre de todos los empleados que viven en la misma ciudad y en la misma calle que sus jefes.
  - b. Determinar el nombre de todos los empleados de esta base de datos que no trabajan para el Banco Importante.
  - c. Determinar el nombre de todos los empleados que ganan más que cualquier empleado del Banco Pequeño.
- 2.2 Las operaciones de reunión externa amplían la operación reunión natural de manera que las tuplas de las relaciones participantes no se pierdan en el resultado de la reunión. Descríbase la manera en que la operación reunión zeta puede ampliarse para que las tuplas de la relación de la izquierda, las de la relación de la derecha o las de ambas relaciones no se pierdan en el resultado de las reuniones zeta.
- 2.3 Considérese la base de datos relacional de la Figura 2.35. Obténgase una expresión del álgebra relacional para cada una de las peticiones siguientes:
- a. Modificar la base de datos de manera que Santos viva ahora en Tres Cantos.
  - b. Dar a todos los jefes de la base de datos un aumento de sueldo del 10 por ciento.

## Ejercicios

- 2.4 Descríbanse las diferencias de significado entre los términos *relación* y *esquema de la relación*.

*empleado* (nombre\_persona, calle, ciudad)  
*trabaja* (nombre\_persona, nombre\_empresa, sueldo)  
*empresa* (nombre\_empresa, ciudad)  
*jefe* (nombre\_persona, nombre\_jefe)

**Figura 2.35** Base de datos relacional para los ejercicios 2.1, 2.3, 2.5, 2.7 y 2.9.



- 2.5 Considérese la base de datos relacional de la Figura 2.35, en la que las claves primarias están subrayadas. Obténgase una expresión del álgebra relacional para expresar cada una de las consultas siguientes:
- Determinar los nombres de todos los empleados que trabajan para el Banco Importante.
  - Determinar el nombre y la ciudad de residencia de todos los empleados que trabajan para el Banco Importante.
  - Determinar el nombre, la calle y la ciudad de residencia de todos los empleados que trabajan para el Banco Importante y ganan más de 10.000 € anuales.
  - Determinar el nombre de todos los empleados de esta base de datos que viven en la misma ciudad que la compañía para la que trabajan.
  - Supóngase que las compañías pueden estar instaladas en varias ciudades. Hállense todas las compañías instaladas en cada ciudad en la que está instalado el Banco Pequeño.
- 2.6 Considérese la relación de la Figura 2.20, que muestra el resultado de la consulta “Determinar el nombre de todos los clientes que tienen concedido un préstamo en el banco”. Vuélvase a escribir la consulta para incluir no sólo el nombre, sino también la ciudad de residencia de cada cliente. Obsérvese que ahora el cliente Sotoca ya no aparece en el resultado, aunque en realidad tiene un préstamo concedido por el banco.
- Explíquese el motivo de que Sotoca no aparezca en el resultado.
  - Supóngase que se desea que Sotoca aparezca en el resultado. ¿Cómo habría que modificar la base de datos para conseguirlo?
  - Una vez más, supóngase que se desea que Sotoca aparezca en el resultado. Escríbase una consulta que utilice una reunión externa que cumpla esta condición sin que haya que modificar la base de datos.
- 2.7 Considérese la base de datos relacional de la Figura 2.35. Obténgase una expresión del álgebra relacional para cada petición:
- Dar a todos los empleados del Banco Importante un aumento de sueldo del 10 por ciento.
  - Dar a todos los jefes de la base de datos un aumento de sueldo del 10 por ciento, a menos que el sueldo resultante sea mayor que 100.000 €. En ese caso, dar sólo un aumento del 3 por ciento.
  - Borrar todas las tuplas de la relación *trabajo* de los empleados de Banco Pequeño.
- 2.8 Usando el ejemplo bancario, escríbanse consultas del álgebra relacional para encontrar las cuentas abiertas por más de dos clientes:
- Usando una función de agregación.
  - Sin usar funciones de agregación.
- 2.9 Considérese la base de datos relacional de la Figura 2.35. Obténgase una expresión del álgebra relacional para cada una de las consultas siguientes:
- Determinar la compañía con mayor número de empleados.
  - Determinar la compañía con la nómina más reducida.
  - Determinar las compañías cuyos empleados ganen un sueldo más elevado, en media, que el sueldo medio del Banco Importante.
- 2.10 Determinénse dos motivos por los que se puedan introducir valores nulos en la base de datos.
- 2.11 Considérese el siguiente esquema de relación:

*empleado*(número\_empleado, nombre, sucursal, edad)  
*libros*(isbn, título, autores, editorial)  
*préstamo*(número\_empleado, isbn, fecha)

Escríbanse las consultas siguientes en el álgebra relacional.

- Determinar el nombre de los empleados que han tomado prestados libros editados por McGraw-Hill.