

8

capítulo 8

El álgebra relacional y Cálculo relacional

En

el capítulo 8 describimos los lenguajes formales para el cálculo relacional. Por el contrario, los capítulos 6 y 7 describen el lenguaje práctico para el modelo relacional, es decir, el estándar SQL. Históricamente, la álgebra relacional y cálculo se desarrollaron antes que el lenguaje SQL. SQL se basa principalmente sobre conceptos del cálculo relacional y se ha ampliado para incorporar algunos conceptos del álgebra relacional también. Porque la mayoría de DBMS relacionales usan SQL como su lenguaje, presentamos primero el lenguaje SQL.

Recuerde del Capítulo 2 que un modelo de datos debe incluir un conjunto de operaciones para manipular la base de datos, además de los conceptos del modelo de datos para definir la estructura y limitaciones de la base de datos. Presentamos las estructuras y limitaciones del modelo relacional formal en el Capítulo 5. El conjunto básico de operaciones para el modelo relacional formal es el **álgebra relacional**. Estas operaciones permiten a un usuario para especificar solicitudes de recuperación básicas como expresiones de álgebra relacional. El resultado de una consulta de recuperación es una nueva relación. Las operaciones de álgebra producen así nuevas relaciones, que se pueden manipular aún más utilizando operaciones del mismo álgebra. Una secuencia de operaciones de álgebra relacional forma una **expresión de álgebra relacional**, cuyo resultado también será una relación que representa el resultado de una consulta de base de datos (o solicitud de recuperación).

El álgebra relacional es muy importante por varias razones. Primero, proporciona una base formal para las operaciones del modelo relacional. Segundo, y quizás más importante, se utiliza como base para implementar y optimizar consultas en los módulos de optimización y procesamiento de consultas que son parte integral de las relaciones.

sistemas de gestión de bases de datos (RDBMS), como veremos en los capítulos 18 y 19. En tercer lugar, algunos de sus conceptos se incorporan al estándar SQL.

Página 2

lenguaje de consulta para RDBMS. Aunque la mayoría de los RDBMS comerciales que se utilizan actualmente no proporcionan interfaces de usuario para consultas de álgebra relacional, las operaciones centrales y las funciones en los módulos internos de la mayoría de los sistemas relacionales se basan en operaciones de álgebra relacional. Definiremos estas operaciones en detalle en la Sección 8.1 a 8.4 de este capítulo.

Mientras que el álgebra define un conjunto de operaciones para el modelo relacional, el **El cálculo relacional** proporciona un lenguaje declarativo de alto nivel para especificar relaciones consultas nacionales. En una expresión de cálculo relacional, no hay un orden de operaciones para especificar cómo recuperar el resultado de la consulta; solo qué información debe Contiene. Esta es la principal característica que distingue entre álgebra relacional y relacional. El cálculo relacional es importante porque tiene una base firme en lógica matemática y porque el lenguaje de consulta estándar (SQL) para RDBMS tiene algunos de sus fundamentos en una variación del cálculo relacional conocida como tupla cálculo relacional. ¹

El álgebra relacional a menudo se considera una parte integral de los datos relacionales. modelo. Sus operaciones se pueden dividir en dos grupos. Un grupo incluye conjunto de operaciones acciones de la teoría matemática de conjuntos; estos son aplicables porque cada relación es definido como un conjunto de tuplas en el modelo relacional formal (ver Sección 5.1). Conjunto Las operaciones incluyen UNION , INTERSECTION , SET DIFFERENCE y CARTESIAN PRODUCTO (también conocido como PRODUCTO CRUZADO). El otro grupo consiste en operaciones desarrolladas específicamente para bases de datos relacionales, que incluyen SELECT , PROYECTO , y UNIRSE , entre otros. Primero, describimos SELECT y PROJECT operaciones en la Sección 8.1 porque son **operaciones unarias** que operan en un solo relaciones. Luego discutimos las operaciones de conjuntos en la Sección 8.2. En la Sección 8.3, discutimos JOIN y otras **operaciones binarias** complejas , que operan en dos tablas por combinar tuplas relacionadas (registros) según las condiciones de combinación. La EMPRESA relacional La base de datos que se muestra en la Figura 5.6 se utiliza para nuestros ejemplos.

Algunas solicitudes de bases de datos comunes no se pueden realizar con el relacional original. operaciones de álgebra, por lo que se crearon operaciones adicionales para expresar estas solicitudes. Estos incluyen **funciones agregadas** , que son operaciones que pueden resumir datos de las tablas, así como tipos adicionales de operaciones JOIN y UNION , conocidas como OUTER JOIN s y EXTERIOR DE LA UNIÓN s. Estas operaciones, que se agregaron al original álgebra relacional debido a su importancia para muchas aplicaciones de bases de datos, se describen en la Sección 8.4. Damos ejemplos de consultas específicas que utilizan relaciones operaciones nacionales en la Sección 8.5. Algunas de estas mismas consultas se utilizaron en los capítulos 6 y 7. Al utilizar los mismos números de consulta en este capítulo, el lector puede contrastar cómo se escriben las mismas consultas en los distintos lenguajes de consulta.

En las Secciones 8.6 y 8.7 describimos el otro lenguaje formal principal para relacionales bases de datos, el **cálculo relacional**. Hay dos variaciones de cálculo relacional. El cálculo relacional de tuplas se describe en la sección 8.6 y el dominio relacional El cálculo se describe en la Sección 8.7. Algunas de las construcciones SQL discutidas en

¹ SQL se basa en el cálculo relacional de tuplas, pero también incorpora algunas de las operaciones del álgebra relacional y sus extensiones, como se ilustra en los capítulos 6, 7 y 9.

Los capítulos 6 y 7 se basan en el cálculo relacional de tuplas. El cálculo relacional es un lenguaje formal, basado en la rama de la lógica matemática llamada predicado cálculo. ² En el cálculo relacional de tuplas, las variables se extienden sobre tuplas, mientras que en Cálculo relacional de dominios, las variables varían sobre los dominios (valores) de los atributos. En el Apéndice C damos una descripción general del lenguaje Query-By-Example (QBE), que es un lenguaje relacional gráfico fácil de usar basado en el dominio relacional cálculo. La sección 8.8 resume el capítulo.

Para el lector que esté interesado en una introducción menos detallada a las relaciones formales idiomas, las Secciones 8.4, 8.6 y 8.7 pueden omitirse.

8.1 Operaciones relacionales unarias: SELECCIONAR y PROYECTAR

8.1.1 La operación SELECT

La operación SELECT se usa para elegir un subconjunto de las tuplas de una relación que satisface una **condición de selección**. ³ Podemos considerar la operación SELECT como un filtro que conserva solo las tuplas que satisfacen una condición de calificación. Alternativamente, podemos considere la operación SELECT para restringir las tuplas en una relación solo a esas tuplas que satisfacen la condición. La operación SELECT también se puede visualizar como un horizonte partición tal de la relación en dos conjuntos de tuplas, aquellas tuplas que satisfacen la condición y se seleccionan, y aquellas tuplas que no satisfacen la condición y son filtrado. Por ejemplo, para seleccionar las tuplas EMPLEADO cuyo departamento es 4, o aquellos cuyo salario sea superior a \$ 30.000, podemos especificar individualmente cada uno de estos dos condiciones con una operación SELECT como sigue:

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

$$\sigma_{Salario > 30000} (EMPLEADO)$$

En general, la operación SELECT se denota por

$$\sigma_{< \text{condición de selección} >} (R)$$

donde el símbolo σ (sigma) se usa para denotar el operador SELECT y la selección condición es una expresión booleana (condición) especificada en los atributos de

relación R. Note que R es generalmente una expresión de álgebra relacional cuyo resultado es un relación: la expresión más simple de este tipo es simplemente el nombre de una relación de base de datos. Los La relación resultante de la operación SELECT tiene los mismos atributos que R.

La expresión booleana especificada en <condición de selección> se compone de un número de **cláusulas** de la forma

<nombre de atributo> <operación de comparación> <valor constante>

² En este capítulo no estoy familiarizado con el cálculo de predicados de primer orden, que se ocupa de las variables cuantificadas. y valores — se asume.

³ La operación SELECT es diferente de la cláusula SELECT de SQL. La operación SELECT elige tuplas de una tabla, y a veces se denomina operación RESTRICT o FILTER.

Página 4

242 Capítulo 8 El álgebra relacional y el cálculo relacional

O

<nombre de atributo> <operación de comparación> <nombre de atributo>

donde <nombre de atributo> es el nombre de un atributo de R, <operación de comparación> es normalmente uno de los operadores {=, <, ≤, >, ≥, ≠} y <valor constante> es una constante valor del dominio de atributo. Las cláusulas se pueden conectar mediante el estándar booleano operadores y, o, y no para formar una condición de selección general. Por ejemplo, para seleccione las tuplas para todos los empleados que trabajan en el departamento 4 y se renuevan \$ 25,000 por año, o trabajar en el departamento 5 y ganar más de \$ 30,000, podemos especificar la siguiente operación SELECT :

$\sigma (Dno = 4 \vee Salario > 25000) \cup \sigma (Dno = 5 \vee Salario > 30000) (EMPLEADO)$

El resultado se muestra en la Figura 8.1 (a).

Observe que todos los operadores de comparación del conjunto {=, <, ≤, >, ≥, ≠} pueden aplicarse a atributos cuyos dominios son valores ordenados, como dominios numéricos o de fecha. Los dominios de cadenas de caracteres también se consideran ordenados en función de la secuencia de los personajes. Si el dominio de un atributo es un conjunto de valores, entonces solo se pueden usar los operadores de comparación en el conjunto {=, ≠}. Un ejemplo de un dominio desordenado es el dominio Color = {'rojo', 'azul', 'verde', 'blanco', 'amarillo', ...}, donde no se especifica ningún orden entre los distintos colores. Algunos dominios permitir tipos adicionales de operadores de comparación; por ejemplo, un dominio de carácter Las cadenas pueden permitir al operador de comparación SUBSTRING_OF .

Figura 8.1

Resultados de las operaciones SELECT y PROJECT. (a) $\sigma (Dno = 4 \vee Salario > 25000) \cup \sigma (Dno = 5 \vee Salario > 30000) (EMPLEADO)$.

(b) $\pi Lnombre, Fnombre, Salario (EMPLEADO)$. (c) $\pi Sexo, Salario (EMPLEADO)$.

(un)

Fname	Minit	Lname	Ssn	Bdate	Habla a	Salario sexual Super_ssn Dno
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	METR00000 888665555 5
Jennifer	S	Wallace	987654321	20/06/1941	291 Berry, Bellaire, TX	F 43000 888665555 4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	METR00000 333445555 5

(segundo)

Lname	Fname	Salario	Sexo	Salario
Herrero	Juan	30000	METR00000	
Wong	Franklin	40000	METR00000	
Zelaya	Alicia	25000	F	25000
Wallace	Jennifer	43000	F	43000
Narayan	Ramesh	38000	METR08000	
Inglés	Joyce	25000	METR05000	
Jabbar	Ahmad	25000	METR05000	
Borg	James	55000		

(C)

Página 5

8.1 Operaciones relacionales unarias: SELECT y PROJECT 243

En general, el resultado de una operación SELECT se puede determinar como sigue. los <condición de selección> se aplica independientemente a cada tupla individual t en R . Este se realiza sustituyendo cada aparición de un atributo A_i en la condición de selección con su valor en la tupla t $[A_i]$. Si la condición se evalúa como VERDADERA, entonces la tupla t es **seleccionado**. Todas las tuplas seleccionadas aparecen en el resultado de la operación SELECT. los Las condiciones booleanas \vee , \wedge y NO tienen su interpretación normal, de la siguiente manera:

- $(\text{cond1} \wedge \text{cond2})$ es TRUE si tanto (cond1) como (cond2) son TRUE; otro-sabio, es FALSO.
- $(\text{cond1} \vee \text{cond2})$ es TRUE si (cond1) o (cond2) o ambos son TRUE; de lo contrario, es FALSO.
- (NOT cond) es TRUE si cond es FALSE; de lo contrario, es FALSO.

El operador SELECT es **unario**; es decir, se aplica a una sola relación. Además, la operación de selección se aplica a cada tupla individualmente; por tanto, la condición de selección Las descripciones no pueden involucrar más de una tupla. El **grado** de la relación resultante de una operación SELECT (su número de atributos) es el mismo que el grado de R . número de tuplas en la relación resultante es siempre menor o igual que el número de tuplas en R . Es decir, $|\sigma(R)| \leq |R|$ para cualquier condición C . La fracción de tuplas seleccionado mediante una condición de selección se denomina **selectividad** de la condición.

Observe que la operación SELECT es **conmutativa**; es decir,

$$\sigma_{\langle \text{cond1} \rangle}(\sigma_{\langle \text{cond2} \rangle}(R)) = \sigma_{\langle \text{cond2} \rangle}(\sigma_{\langle \text{cond1} \rangle}(R))$$

Por tanto, se puede aplicar una secuencia de SELECT s en cualquier orden. Además, podemos Siempre combine una **cascada** (o **secuencia**) de operaciones SELECT en una sola SELECT

operación con una condición conjuntiva (\wedge); es decir,

$$\sigma_{< \text{cond1} >} (\sigma_{< \text{cond2} >} (\dots (\sigma_{< \text{condn} >} (R)) \dots)) = \sigma_{< \text{cond1} >} \wedge \sigma_{< \text{cond2} >} \wedge \dots \wedge \sigma_{< \text{condn} >} (R)$$

En SQL, la condición SELECT generalmente se especifica en la cláusula WHERE de una consulta. Por ejemplo, la siguiente operación:

$\sigma_{\text{Dno} = 4 \wedge \text{Salario} > 25000} (\text{EMPLEADO})$

correspondería a la siguiente consulta SQL:

```
SELECCIONE *
DESDE      EMPLEADO
DÓNDE      Dno = 4 Y Salario > 25000 ;
```

8.1.2 La operación PROYECTO

Si pensamos en una relación como una tabla, la operación SELECT elige algunas de las filas de la tabla mientras descarta otras filas. La operación PROYECTO, por otro lado, selecciona ciertas columnas de la tabla y descarta las otras columnas. Si nosotros estamos interesados solo en ciertos atributos de una relación, usamos la operación PROYECTO para proyectar la relación sobre estos atributos solamente. Por tanto, el resultado del PROYECTO La operación se puede visualizar como una partición vertical de la relación en dos relaciones:

uno tiene las columnas necesarias (atributos) y contiene el resultado de la operación, y el otro contiene las columnas descartadas. Por ejemplo, para enumerar los nombre y apellido y salario, podemos utilizar la operación PROYECTO de la siguiente manera:

$\pi_{\text{Lname, Fname, Salario}} (\text{EMPLEADO})$

La relación resultante se muestra en la Figura 8.1 (b). La forma general del PROYECTO la operación es

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

donde π (pi) es el símbolo utilizado para representar la operación PROYECTO, y $< \text{atributo lista} >$ es la sublista deseada de atributos de los atributos de la relación R. De nuevo, observe que R es, en general, una expresión de álgebra relacional cuyo resultado es una relación, que en el caso más simple es solo el nombre de una relación de base de datos. El resultado de la La operación PROYECTO tiene solo los atributos especificados en $< \text{lista de atributos} >$ en el mismo orden como aparecen en la lista. Por tanto, su **grado** es igual al número de atributos en $< \text{lista de atributos} >$.

Si la lista de atributos incluye solo atributos no clave de R, las tuplas duplicadas son probable que ocurra. La operación PROYECTO elimina cualquier tupla duplicada, por lo que El resultado de la operación PROYECTO es un conjunto de tuplas distintas y, por lo tanto, una relación. Esto se conoce como **eliminación de duplicados**. Por ejemplo, considere el

siguiente operación de PROYECTO :

$$\pi_{\text{Sexo, salario}} (\text{EMPLEADO})$$

El resultado se muestra en la Figura 8.1 (c). Observe que la tupla <'F', 25000> aparece solo una vez en la Figura 8.1 (c), aunque esta combinación de valores aparece dos veces en la Relación EMPLEADO. La eliminación de duplicados implica la clasificación o alguna otra técnica para detecta duplicados y, por lo tanto, agrega más procesamiento. Si no se eliminan los duplicados, el resultado sería un conjunto **múltiple** o una **bolsa** de tuplas en lugar de un conjunto. Esto no estaba permitido en el modelo relacional formal pero está permitido en SQL (vea la Sección 6.3).

El número de tuplas en una relación resultante de una operación PROYECTO es siempre menor o igual que el número de tuplas en R. Si la lista de proyección es una superclave de R, es decir, incluye alguna clave de R, la relación resultante tiene el mismo número de tuplas como R. Además,

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

siempre que <list2> contenga los atributos de <list1>; de lo contrario, el lado izquierdo es una expresión incorrecta. También es digno de mención que la conmutatividad no se sostiene en PROYECTO.

En SQL, la lista de atributos PROJECT se especifica en la cláusula SELECT de una consulta. por ejemplo, la siguiente operación:

$$\pi_{\text{Sexo, salario}} (\text{EMPLEADO})$$

correspondería a la siguiente consulta SQL:

```
SELECCIONE  DISTINTO Sexo, Salario
DESDE      EMPLEADO
```

Observe que si eliminamos la palabra clave DISTINCT de esta consulta SQL, entonces duplicados no se eliminarán los datos. Esta opción no está disponible en el álgebra relacional formal, pero el álgebra se puede ampliar para incluir esta operación y permitir la relación opciones para ser multijuegos; no discutimos estas extensiones aquí.

8.1.3 Secuencias de operaciones y operación RENAME

Las relaciones que se muestran en la Figura 8.1 que representan los resultados de la operación no tienen ninguna nombres. En general, para la mayoría de consultas, necesitamos aplicar varios álgebra relacional operaciones una tras otra. O podemos escribir las operaciones como un solo **expresión de álgebra relacional** anidando las operaciones, o podemos aplicar una operación a la vez y crear relaciones de resultado intermedias. En el último caso, debemos dar nombres a las relaciones que contienen los resultados intermedios. Por ejemplo, para recuperar el nombre, apellido y salario de todos los empleados que trabajan en el departamento número 5, debemos aplicar una operación SELECT y una PROJECT. Podemos escribir un pecado

gle expresión de álgebra relacional, también conocida como **expresión en línea** , como sigue:
 $\pi_{Fname, Lname, Salario}(\sigma_{Dno = 5}(EMPLEADO))$

La figura 8.2 (a) muestra el resultado de esta expresión de álgebra relacional en línea. Alternativamente, podemos mostrar explícitamente la secuencia de operaciones, dando un nombre a cada relación intermedia, y usando la **operación de asignación**, denotada por \leftarrow (izquierda flecha), como sigue:

$DEP5_EMPS \leftarrow \sigma_{Dno = 5}(EMPLEADO)$
 $RESULTADO \leftarrow \pi_{Fname, Lname, Salario}(DEP5_EMPS)$

A veces es más sencillo dividir una secuencia compleja de operaciones especificando relaciones de resultado intermedias que escribir una sola expresión de álgebra relacional. También podemos utilizar esta técnica para **cambiar el nombre de** los atributos en el intermedio y relaciones de resultado. Esto puede ser útil en conexión con operaciones más complejas, como UNION y JOIN , como veremos. Para cambiar el nombre de los atributos en una relación, simplemente enumere los nombres de los nuevos atributos entre paréntesis, como en el siguiente ejemplo:

$TEMP \leftarrow \sigma_{Dno = 5}(EMPLEADO)$
 $R(Nombre, Apellido, Salario) \leftarrow \pi_{Fnombre, Lnombre, Salario}(TEMP)$

Estas dos operaciones se ilustran en la Figura 8.2 (b).

Si no se aplica un cambio de nombre, los nombres de los atributos en la relación resultante de un Las operaciones SELECT son las mismas que las de la relación original y en la misma orden. Para una operación PROYECTO sin cambio de nombre, la relación resultante tiene la mismos nombres de atributo que los de la lista de proyección y en el mismo orden en que aparecen en la lista.

También podemos definir una operación RENAME formal , que puede cambiar el nombre del nombre de la opción o los nombres de los atributos, o ambos, como un operador unario. El general La operación RENAME cuando se aplica a una relación R de grado n se denota por cualquiera de los siguientes tres formas:

$\rho_s(B_1, B_2, \dots, B_n)(R) \text{ o } \rho_s(R) \text{ o } \rho(B_1, B_2, \dots, B_n)(R)$

(un)

Fname	Lname	Salario
Juan	Herrero	30000
Franklin	Wong	40000
Ramesh	Narayan	38000
Joyce	Inglés	25000

(segundo)

TEMPERATURA

Fname	Minit	Lname	Ssn	Bdate	Habla a	Sexo	Salario	Super_ssn	Dno
Juan	segund	Herrero	123456789	1965-01-09	731 Fondren, Houston, TX	METRO	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	METRO	40000	888665555	5

Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	METRO	38000	333445555	5
Joyce	UN	Inglés	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5

R		
Nombre	Apellido	Salario
Juan	Herrero	30000
Franklin	Wong	40000
Ramesh	Narayan	38000
Joyce	Inglés	25000

Figura 8.2
Resultados de una secuencia de operaciones. (a) $\pi_{Fnombre, Lnombre, Salario}(\sigma_{Dno = 5}(EMPLEADO))$.
(b) Uso de relaciones intermedias y cambio de nombre de atributos.

donde el símbolo ρ (rho) se usa para denotar el operador RENAME, S es la nueva relación nombre de la opción, y B_1, B_2, \dots, B_n son los nuevos nombres de atributo. La primera expresión cambia el nombre tanto de la relación como de sus atributos, el segundo sólo cambia el nombre de la relación, y el tercero cambia el nombre de los atributos únicamente. Si los atributos de R son (A_1, A_2, \dots, A_n) en ese orden, entonces cada A_i se renombra como B_i .

En SQL, una sola consulta generalmente representa una expresión de álgebra relacional compleja. El cambio de nombre en SQL se logra mediante la creación de alias mediante AS, como en el siguiente ejemplo:

```
SELECCIONE  E.Fname AS Nombre, E.Lname AS Apellido, E.Salario AS Salario
DESDE       EMPLEADO COMO E
DÓNDE       E.Dno = 5,
```

8.2 Operaciones de álgebra relacional de la teoría de conjuntos

8.2.1 Las operaciones UNION, INTERSECTION y MINUS

El siguiente grupo de operaciones de álgebra relacional son las matemáticas estándar. operaciones en conjuntos. Por ejemplo, para recuperar los números de seguro social de todos

empleados que trabajan en el departamento 5 o supervisan directamente a un empleado que trabaja en el departamento 5, podemos utilizar la operación UNION de la siguiente manera: 4

```
DEP5_EMPS ←  $\sigma_{Dno = 5}(EMPLEADO)$ 
RESULT1 ←  $\pi_{Ssn}(DEP5\_EMPS)$ 
RESULT2 ( $Ssn$ ) ←  $\pi_{Super\_ssn}(DEP5\_EMPS)$ 
RESULTADO ← RESULT1 U RESULT2
```

La relación RESULT1 tiene el Ssn de todos los empleados que laboran en el departamento 5. Considerando que RESULT2 tiene el Ssn de todos los empleados que supervisan directamente a un empleado que trabaja en el departamento 5. La operación UNION produce las tuplas que están en ya sea RESULT1 o RESULT2 o ambos (véase la Figura 8.3) al tiempo que elimina cualquier duplicación cates. Por lo tanto, el valor Ssn '333445555' aparece solo una vez en el resultado.

Se utilizan varias operaciones de la teoría de conjuntos para fusionar los elementos de dos conjuntos en formas, incluyendo UNION , INTERSECTION y SET DIFFERENCE (también llamado

MENOS o EXCEPTO). Estas son operaciones **binarias** ; es decir, cada uno se aplica a dos conjuntos (de tuplas). Cuando estas operaciones se adaptan a las bases de datos relacionales, las dos sobre las que se aplique cualquiera de estas tres operaciones deben tener el mismo **tipo de tuplas** ; esta condición se ha denominado compatibilidad de unión o compatibilidad de tipos.

Dos relaciones $R(A_1, A_2, \dots, A_n)$ y $S(B_1, B_2, \dots, B_n)$ se dice que son **unión compatible** (o **compatible de tipo**) si tienen el mismo grado n y si $\text{dom}(A_i) = \text{dom}(B_i)$ para $1 \leq i \leq n$. Esto significa que las dos relaciones tienen el mismo número de atributos y cada par de atributos correspondiente tiene el mismo dominio.

Podemos definir las tres operaciones UNION , INTERSECTION y SET DIFFERENCE en dos relaciones R y S compatibles con la unión como sigue:

- **UNIÓN** : El resultado de esta operación, denotado por $R \cup S$, es una relación que incluye todas las tuplas que están en R o en S o en R y S. Duplicar se eliminan las tuplas.
- **INTERSECCIÓN** : El resultado de esta operación, denotado por $R \cap S$, es una relación que incluye todas las tuplas que están tanto en R como en S.
- **SET DIFFERENCE (o MINUS)** : el resultado de esta operación, denotado por $R - S$, es una relación que incluye todas las tuplas que están en R pero no en S.

RESULTADO1	RESULTADO2	RESULTADO
Ssn	Ssn	Ssn
123456789	333445555	123456789
333445555	888665555	333445555
666884444		666884444
453453453		453453453
		888665555

Figura 8.3
Resultado de la operación UNION
 $\text{RESULTADO} \leftarrow \text{RESULT1} \cup \text{RESULT2}.$

⁴ Como una sola expresión de álgebra relacional, esto se convierte en $\text{Resultado} \leftarrow \pi_{\text{Ssn}}(\sigma_{\text{Dno}=5}(\text{EMPLEADO})) \cup \pi_{\text{Super_ssn}}(\sigma_{\text{Dno}=5}(\text{EMPLEADO}))$.

resultado utilizando el operador de cambio de nombre.
La figura 8.4 ilustra las tres operaciones. Las relaciones ESTUDIANTE e INSTRUCTOR en la Figura 8.4 (a) son compatibles con las uniones y sus tuplas representan los nombres de los abolladuras y los nombres de los instructores, respectivamente. El resultado de la operación UNION en la Figura 8.4 (b) muestra los nombres de todos los estudiantes e instructores. Tenga en cuenta que duplicar las tuplas aparecen solo una vez en el resultado. El resultado de la operación INTERSECCIÓN (Figura 8.4 (c)) incluye solo a aquellos que son estudiantes e instructores.

Observe que tanto UNION como INTERSECTION son operaciones conmutativas; es decir,

$$R \cup S = S \cup R \text{ y } R \cap S = S \cap R$$

Tanto UNION como INTERSECTION pueden tratarse como operaciones n-arias aplicables a cualquier número de relaciones porque ambas son también operaciones asociativas; es decir,

$$R \cup (S \cup T) = (R \cup S) \cup T \text{ y } (R \cap S) \cap T = R \cap (S \cap T)$$

Figura 8.4
El conjunto de operaciones UNIÓN, INTERSECCIÓN y MENOS. (a) Dos relaciones sindicalmente compatibles. (b) ESTUDIANTE \cup INSTRUCTOR. (c) ESTUDIANTE \cap INSTRUCTOR. (d) ESTUDIANTE - INSTRUCTOR. (e) INSTRUCTOR - ESTUDIANTE.

(un) ESTUDIANTE		INSTRUCTOR		(segundo)	
F _n	L _n	F _{name}	L _{name}	F _n	L _n
Susan	Yao	Juan	Herrero	Susan	Yao
Ramesh	Cha	Ricardo	Browne	Ramesh	Cha
Johnny	Kohler	Susan	Yao	Johnny	Kohler
Bárbara	Jones	Francis	Johnson	Bárbara	Jones
Amy	Vado	Ramesh	Cha	Amy	Vado
Palanqueta	Wang			Palanqueta	Wang
Ernesto	Gilbert			Ernesto	Gilbert
				Juan	Herrero
				Ricardo	Browne
				Francis	Johnson

(C)		(re)		(mi)	
F _n	L _n	F _n	L _n	F _{name}	L _{name}
Susan	Yao	Johnny	Kohler	Juan	Herrero
Ramesh	Cha	Bárbara	Jones	Ricardo	Browne
		Amy	Vado	Francis	Johnson
		Palanqueta	Wang		
		Ernesto	Gilbert		

La operación MINUS no es conmutativa; es decir, en general,

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

La figura 8.4 (d) muestra los nombres de los estudiantes que no son profesores, y la figura 8.4 (e) muestra los nombres de los instructores que no son estudiantes.

Tenga en cuenta que la INTERSECCIÓN se puede expresar en términos de unión y establecer la diferencia como sigue:

$$R \cap S = ((R \cup S) - (R - S)) - (S - R)$$

En SQL, hay tres operaciones: UNION, INTERSECT y EXCEPT, que corresponden a las operaciones establecidas aquí descritas. Además, existen operaciones multiset (UNION ALL, INTERSECT ALL y EXCEPT ALL) que no eliminan duplicados (consulte la Sección 6.3.4).

8.2.2 El PRODUCTO CARTESIANO (PRODUCTO CRUZADO)

Operación

A continuación, analizamos la operación del PRODUCTO CARTESIANO, también conocida como CROSS PRODUCTO o UNIÓN CRUZADA, que se indica con \times . Esta también es una operación de conjunto binario, pero las relaciones sobre las que se aplica no tienen que ser compatibles sindicalmente. En su forma binaria, esta operación de conjunto produce un nuevo elemento combinando cada miembro (tupla) de una relación (conjunto) con cada miembro (tupla) de la otra relación (conjunto). En general, el resultado de $R(A_1, A_2, \dots, A_n) \times S(B_1, B_2, \dots, B_m)$ es una relación Q con atributos de grado $n + m$ $Q(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$, en ese orden. La relación resultante Q tiene una tupla para cada combinación de tuplas, una de R y uno de S . Por lo tanto, si R tiene n_R tuplas (denotadas como $|R| = n_R$), y S tiene n_S tuplas, entonces $R \times S$ tendrá $n_R * n_S$ tuplas.

La operación del PRODUCTO CARTESIANO n -ario es una extensión del concepto anterior, que produce nuevas tuplas concatenando todas las combinaciones posibles de tuplas de n relaciones subyacentes. La operación PRODUCTO CARTESIANO aplicada por sí misma generalmente no tiene sentido. Es sobre todo útil cuando va seguido de una selección que coincide con los valores de los atributos que provienen de las relaciones de los componentes. Por ejemplo, supongamos que queremos recuperar una lista de nombres de las dependencias de cada empleada abolladuras. Podemos hacer esto de la siguiente manera:

```
FEMALE_EMPS ← σ Sexo = 'F' ( EMPLEADO )
EMP_NAMES ← π Fname, Lname, Ssn ( FEMALE_EMPS )
EMP_DEPENDENTS ← EMP_NAMES × DEPENDENT
ACTUAL_DEPENDENTS ← σ Ssn = Essn ( EMP_DEPENDENTS )
RESULTADO ← π Fname, Lname, Dependent_name ( ACTUAL_DEPENDENTS )
```

Las relaciones resultantes de esta secuencia de operaciones se muestran en la Figura 8.5.

La relación EMP_DEPENDENTS es el resultado de aplicar el PRODUCTO CARTESIANO operación a EMP_NAMES de la Figura 8.5 con DEPENDENT de la Figura 5.6. En

EMP_DEPENDENTS, cada tupla de EMP_NAMES se combina con cada tupla de DEPENDIENTE, dando un resultado que no es muy significativo (cada dependiente es

Figura 8.5

La operación PRODUCTO CARTESIANO (PRODUCTO CRUZADO).

FEMALE_EMPS

Fname	Minit	Lname	Ssn	Bdate	Habla a	Sexo	Salario	Super_ssn	Dno
Alicia	J	Zelaya	999887777	1968-07-19	3321 Castillo, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	20/06/1941	291Berry, Bellaire, TX	F	43000	888665555	4
Joyce	UN	Inglés	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5

EMPNames

Fname	Lname	Ssn
Alicia	Zelaya	999887777
Jennifer	Wallace	987654321
Joyce	Inglés	453453453

EMP_DEPENDENTS

Fname	Lname	Ssn	Essn	Nombre_dependiente	Sexo	Bdate	...
Alicia	Zelaya	999887777	333445555	Alicia	F	1986-04-05	...
Alicia	Zelaya	999887777	333445555	Theodore	M	1983-10-25	...
Alicia	Zelaya	999887777	333445555	Alegría	F	1958-05-03	...
Alicia	Zelaya	999887777	987654321	Abner	M	1942-02-28	...
Alicia	Zelaya	999887777	123456789	Miguel	M	1988-01-04	...
Alicia	Zelaya	999887777	123456789	Alicia	F	1988-12-30	...
Alicia	Zelaya	999887777	123456789	Elizabeth	F	05/05/1967	...
Jennifer	Wallace	987654321	333445555	Alicia	F	1986-04-05	...
Jennifer	Wallace	987654321	333445555	Theodore	M	1983-10-25	...
Jennifer	Wallace	987654321	333445555	Alegría	F	1958-05-03	...
Jennifer	Wallace	987654321	987654321	Abner	M	1942-02-28	...
Jennifer	Wallace	987654321	123456789	Miguel	M	1988-01-04	...
Jennifer	Wallace	987654321	123456789	Alicia	F	1988-12-30	...
Jennifer	Wallace	987654321	123456789	Elizabeth	F	05/05/1967	...
Joyce	Inglés	453453453	333445555	Alicia	F	1986-04-05	...
Joyce	Inglés	453453453	333445555	Theodore	M	1983-10-25	...
Joyce	Inglés	453453453	333445555	Alegría	F	1958-05-03	...
Joyce	Inglés	453453453	987654321	Abner	M	1942-02-28	...
Joyce	Inglés	453453453	123456789	Miguel	M	1988-01-04	...
Joyce	Inglés	453453453	123456789	Alicia	F	1988-12-30	...
Joyce	Inglés	453453453	123456789	Elizabeth	F	05/05/1967	...

ACTUAL_DEPENDENTS

Fname	Lname	Ssn	Essn	Nombre_dependiente	Sexo	Bdate	...
Jennifer	Wallace	987654321	987654321	Abner	M	1942-02-28	...

RESULTADO

Fname	Lname	nombre_dependiente
Jennifer	Wallace	Abner

combinado con todas las empleadas). Queremos combinar una empleada tupla sólo con sus dependientes particulares, es decir, las tuplas **DEPENDIENTES** cuyas El valor de *Essn* coincide con el valor de *Ssn* de la tupla **EMPLOYEE**. Los **ACTUAL_DEPENDENTS** la relación logra esto. La relación **EMP_DEPENDENTS** es un buen ejemplo de el caso donde el álgebra relacional se puede aplicar correctamente para producir resultados que hacen no tiene ningún sentido. Es responsabilidad del usuario asegurarse de aplicar solo significados operaciones importantes a las relaciones.

El **PRODUCTO CARTESIANO** crea tuplas con los atributos combinados de dos relaciones ciones. Podemos **SELECCIONAR** tuplas relacionadas solo de las dos relaciones especificando un condición de selección apropiada después del producto cartesiano, como hicimos en el cediendo ejemplo. Porque esta secuencia de **PRODUCTO CARTESIANO** seguida de **SELECT** se usa con bastante frecuencia para combinar tuplas relacionadas de dos relaciones, una La operación especial, llamada **JOIN**, fue creada para especificar esta secuencia como una sola operación. ción. A continuación, hablamos de la operación **JOIN**.

En **SQL**, **PRODUCTO CARTESIANO** se puede realizar usando la opción **CROSS JOIN** en tablas unidas (consulte la Sección 7.1.6). Alternativamente, si hay dos tablas en el **FROM** cláusula y no hay una condición de unión correspondiente en la cláusula **WHERE** de la Consulta **SQL**, el resultado también será el **PRODUCTO CARTESIANO** de las dos tablas (ver Q10 en la Sección 6.3.3).

8.3 Operaciones relacionales binarias: ÚNETE y DIVISIÓN

8.3.1 La operación JOIN

La operación **JOIN**, denotada por, se usa para combinar tuplas relacionadas de dos relaciones ciones en tuplas simples "más largas". Esta operación es muy importante para cualquier relación base de datos con más de una relación porque nos permite procesar la relación naves entre parientes. Para ilustrar **JOIN**, suponga que queremos recuperar el nombre del gerente de cada departamento. Para obtener el nombre del gerente, necesitamos comparar combinar cada tupla de departamento con la tupla de empleado cuyo valor *Ssn* coincide con el Valor de *Mgr_ssn* en la tupla del departamento. Hacemos esto usando la operación **JOIN** y luego proyectando el resultado sobre los atributos necesarios, de la siguiente manera:

```
DEPT_MGR ← DEPARTMENT Mgr_ssn = Ssn EMPLEADO
RESULTADO ← π Dname, Lname, Fname ( DEPT_MGR )
```

La primera operación se ilustra en la Figura 8.6. Tenga en cuenta que *Mgr_ssn* es una clave externa de la relación **DEPARTAMENTO** que hace referencia a *Ssn*, la clave principal del **EMPLEADO** relación. Esta restricción de integridad referencial juega un papel en tener tuplas coincidentes en la relación de referencia **EMPLEADO**.

La operación **JOIN** se puede especificar como una operación de **PRODUCTO CARTESIANO** siguiendo activado por una operación **SELECT**. Sin embargo, **JOIN** es muy importante porque es se utiliza con frecuencia al especificar consultas de base de datos. Considere el ejemplo anterior

Figura 8.6

Resultado de la operación $\text{JOIN DEPT_MGR} \leftarrow \text{DEPARTMENT} \bowtie_{\text{Mgr_ssn} = \text{Ssn}} \text{EMPLOYEE}$.

DEPT_MGR

Dname	Dnumber	Mgr_ssn	...	Fname	Minit	Lname	Ssn	...
Investigación	5	333445555	...	Franklin	T	Wong	333445555	...
Administración	4	987654321	...	Jennifer	S	Wallace	987654321	...
Sede	1	888665555	...	James	mi	Borg	888665555	...

ilustrando PRODUCTO CARTESIANO, que incluía la siguiente secuencia de operaciones:

$$\begin{aligned}\text{EMP_DEPENDENTS} &\leftarrow \text{EMP_NAMES} \times \text{DEPENDENT} \\ \text{ACTUAL_DEPENDENTS} &\leftarrow \sigma_{\text{Ssn} = \text{Essn}}(\text{EMP_DEPENDENTS})\end{aligned}$$

Estas dos operaciones se pueden reemplazar con una sola operación JOIN de la siguiente manera:

$$\text{ACTUAL_DEPENDENTS} \leftarrow \text{EMP_NAMES} \bowtie_{\text{Ssn} = \text{Essn}} \text{DEPENDENT}$$

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

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

El resultado de JOIN es una relación Q con $n + m$ atributos $Q(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$ en ese orden; Q tiene una tupla para cada combinación de tuplas, una de R y uno de S, siempre que la combinación satisfaga la condición de unión. Esto es la principal diferencia entre PRODUCTO CARTESIANO y UNIRSE. En JOIN, solo combinaciones de tuplas que satisfacen la condición de unión aparecen en el resultado, mientras que en el PRODUCTO CARTESIANO todas las combinaciones de tuplas se incluyen en el resultado. La condición de unión se especifica en atributos de las dos relaciones R y S y es evaluado para cada combinación de tuplas. Cada combinación de tupla para la que la condición de unión se evalúa como VERDADERA se incluye en la relación resultante Q como una tupla combinada.

Una condición de unión general tiene la forma

$$\langle \text{condición} \rangle \ Y \ \langle \text{condición} \rangle \ Y \dots \ Y \ \langle \text{condición} \rangle$$

donde cada $\langle \text{condición} \rangle$ es de la forma $A_i \theta B_j$, A_i es un atributo de R, B_j es un atributo de S, A_i y B_j tienen el mismo dominio, y θ (theta) es uno de los operadores $\{=, <, \leq, >, \geq, \neq\}$. Una operación JOIN con tal condición de unión general es llamado THETA JOIN. Tuplas cuyos atributos de combinación son NULL o para las que la combinación la condición es FALSA no aparecen en el resultado. En ese sentido, la operación JOIN no necesariamente conserva toda la información en las relaciones de participación, porque las tuplas que no se combinan con las que coinciden en la otra relación no aparecen en el resultado.

⁵ Nuevamente, observe que R y S pueden ser cualquier relación que resulte de expresiones de álgebra relacional general.

8.3.2 Variaciones de JOIN: EQUIJOIN y NATURAL JOIN

El uso más común de JOIN implica condiciones de unión con comparaciones de igualdad solamente. Tal JOIN, donde el único operador de comparación utilizado es =, se denomina EQUIJOIN. Ambos ejemplos anteriores fueron EQUIJOINS. Note que en el resultado de un EQUIJOIN siempre tenemos uno o más pares de atributos que tienen valores idénticos en cada tupla. Por ejemplo, en la Figura 8.6, los valores de los atributos Mgr_ssn y Ssn son idénticos en cada tupla de DEPT_MGR (el resultado EQUIJOIN) porque la condición de combinación de igualdad especificada en estos dos atributos requiere que los valores sean idénticos en cada tupla en el resultado. Porque uno de cada par de atributos con valores idénticos es superfluo, una nueva operación llamada NATURAL JOIN —denotada por *— fue creada para eliminar el segundo atributo (superfluo) en un EQUIJOIN condición. La definición estándar de NATURAL JOIN requiere que los dos se unan los atributos (o cada par de atributos de combinación) tienen el mismo nombre en ambas relaciones. Si este no es el caso, primero se aplica una operación de cambio de nombre.

Suponga que queremos combinar cada tupla de PROYECTO con la tupla DEPARTAMENTO que controla el proyecto. En el siguiente ejemplo, primero cambiamos el nombre del atributo Dnumber de DEPARTMENT a Dnum, de modo que tenga el mismo nombre que el atributo Dnum en PROYECTO —y luego aplicamos NATURAL JOIN:

```
PROJ_DEPT ← PROJECT * ρ ( Dname, Dnum, Mgr_ssn, Mgr_start_date ) ( DEPARTAMENTO )
```

La misma consulta se puede realizar en dos pasos creando una tabla intermedia DEPT como sigue:

```
DEPT ← ρ ( Dname, Dnum, Mgr_ssn, Mgr_start_date ) ( DEPARTAMENTO )
PROJ_DEPT ← PROYECTO * DEPT
```

El atributo Dnum se denomina **atributo de unión** para la operación NATURAL JOIN, porque es el único atributo con el mismo nombre en ambas relaciones. La resultante La relación se ilustra en la Figura 8.7 (a). En la relación PROJ_DEPT, cada tupla combina una tupla de PROYECTO con la tupla DEPARTAMENTO para el departamento que controla el proyecto, pero solo se conserva un valor de atributo de unión.

Si los atributos en los que se especifica la combinación natural ya tienen los mismos nombres en ambas relaciones, el cambio de nombre es innecesario. Por ejemplo, para aplicar una combinación natural en los atributos Dnumber de DEPARTMENT y DEPT_LOCATIONS, es suficiente escribir

```
DEPT_LOCS ← DEPARTMENT * DEPT_LOCATIONS
```

La relación resultante se muestra en la Figura 8.7 (b), que combina cada departamento con sus ubicaciones y tiene una tupla para cada ubicación. En general, la condición de unión para NATURAL JOIN se construye equiparando cada par de atributos de combinación que tienen el mismo nombre en las dos relaciones y combinando estas condiciones con AND. Puede haber una lista de atributos de combinación de cada relación, y cada correspondiente el par debe tener el mismo nombre.

⁶ NATURAL JOIN es básicamente un EQUIJOIN seguido de la eliminación de los atributos superfluos.

Página 16

254 Capítulo 8 El álgebra relacional y el cálculo relacional

(un)

PROJ_DEPT

Pname	Pnumber	Ubicación	Dnum	Dname	Mgr_ssn	Mgr_start_date
ProductoX	1	Bellaire	5	Investigación	333445555	1988-05-22
ProductoY	2	La tierra del azúcar	5	Investigación	333445555	1988-05-22
ProductoZ	3	Houston	5	Investigación	333445555	1988-05-22
Informatización	10	Stafford	4	Administración	987654321	1995-01-01
Reorganización	20	Houston	1	Sede	888665555	1981-06-19
Nuevosbeneficios	30	Stafford	4	Administración	987654321	1995-01-01

(segundo)

DEPT_LOCS

Dname	Dnumber	Mgr_ssn	Mgr_start_date	Ubicación
Sede	1	888665555	1981-06-19	Houston
Administración	4	987654321	1995-01-01	Stafford
Investigación	5	333445555	1988-05-22	Bellaire
Investigación	5	333445555	1988-05-22	La tierra del azúcar
Investigación	5	333445555	1988-05-22	Houston

Figura 8.7

Resultados de dos operaciones de unión natural. (a) proj_dept \leftarrow proyecto * dept.(b) dept_locs \leftarrow departamento * dept_locations.

Observe que si ninguna combinación de tuplas satisface la condición de unión, el resultado de una JOIN es una relación vacía con cero tuplas. En general, si R tiene n_R tuplas y S tiene n_S tuplas, el resultado de una operación JOIN R \langle condición de unión \rangle S tendrá entre cero y $n_R * n_S$ tuplas. El tamaño esperado del resultado de la combinación dividido por el máximo el tamaño $n_R * n_S$ conduce a una relación llamada **selectividad de unión**, que es una propiedad de cada unión condición. Si no hay una condición de unión, todas las combinaciones de tuplas califican y el JOIN degenera en un PRODUCTO CARTESIANO, también llamado PRODUCTO CRUZADO o ÚNETE CRUZADO.

Como podemos ver, una sola operación JOIN se usa para combinar datos de dos relaciones de modo que esa información relacionada se puede presentar en una sola tabla. Estas operaciones son también conocidas como **combinaciones internas**, para distinguirlas de una variación de combinación diferente llamada combinaciones externas (consulte la Sección 8.4.4). De manera informal, una combinación interna es un tipo de combinación y operación de combinación definida formalmente como una combinación de PRODUCTO CARTESIANO

SELECCIÓN. Tenga en cuenta que a veces se puede especificar una combinación entre una relación y sí mismo, como haremos en la Sección 8.4.5. El NATURAL JOIN o EQUIJOIN. La operación también se puede especificar entre varias tablas, lo que lleva a una unión de n vías. por Por ejemplo, considere la siguiente combinación de tres vías:

((PROYECTO Dnum = Dnumber DEPARTAMENTO) Mgr_ssn = Ssn EMPLEADO)

8.3 Operaciones relacionales binarias: JOIN y DIVISION 255

Esto combina cada tupla de proyecto con su tupla de departamento de control en una sola tupla, y luego combina esa tupla con una tupla de empleado que es el departamento gerente. El resultado neto es una relación consolidada en la que cada tupla contiene esta información combinada de proyecto-departamento-gerente.

En SQL, JOIN se puede realizar de varias formas diferentes. El primer método es especificar las <condiciones de unión> en la cláusula WHERE , junto con cualquier otra condición de selección. Esto es muy común y se ilustra con las consultas Q1 , Q1A , Q1B , Q2 y Q8 en las Secciones 6.3.1 y 6.3.2, así como por muchos otros ejemplos de consultas en los Capítulos 6 y 7. La segunda forma es utilizar una relación anidada, como se ilustra en las consultas Q4A. y P16 en la Sección 7.1.2. Otra forma es utilizar el concepto de tablas unidas, como ilustrado por las consultas Q1A , Q1B , Q8B y Q2A en la Sección 7.1.6. El constructo de tablas unidas se agregó a SQL2 para permitir al usuario especificar explícitamente todas las varios tipos de combinaciones, porque los otros métodos eran más limitados. También permite el usuario para distinguir claramente las condiciones de unión de las condiciones de selección en el DONDE cláusula.

8.3.3 Un conjunto completo de operaciones de álgebra relacional

Se ha demostrado que el conjunto de operaciones de álgebra relacional $\{\sigma, \pi, \cup, \rho, -, \times\}$ es un juego **completo** ; es decir, cualquiera de las otras operaciones originales de álgebra relacional puede ser expresado como una secuencia de operaciones de este conjunto. Por ejemplo, la INTERSECCIÓN. La operación se puede expresar usando UNION y MINUS de la siguiente manera:

$$R \cap S \equiv (R \cup S) - ((R - S) \cup (S - R))$$

Aunque, estrictamente hablando, no se requiere INTERSECCIÓN , es inconveniente especificar esta expresión compleja cada vez que deseemos especificar una intersección. Como Otro ejemplo, una operación JOIN se puede especificar como un PRODUCTO CARTESIANO siguiendo reducido por una operación SELECT , como comentamos:

$$R \text{ <condición> } S \equiv \sigma \text{ <condición> } (R \times S)$$

De manera similar, un NATURAL JOIN se puede especificar como un PRODUCTO CARTESIANO precedido por RENAME y seguido de las operaciones SELECT y PROJECT . Por lo tanto, los diversos Las operaciones JOIN tampoco son estrictamente necesarias para el poder expresivo de la relación. álgebra nacional. Sin embargo, es importante incluirlos como operaciones independientes. porque son convenientes de usar y se aplican muy comúnmente en bases de datos aplicaciones. Otras operaciones se han incluido en el álgebra relacional básica.

por conveniencia más que por necesidad. Discutimos uno de estos: la DIVISIÓN operación — en la siguiente sección.

8.3.4 La operación DIVISION

La operación DIVISION , denotada por \div , es útil para un tipo especial de consulta que a veces ocurre en aplicaciones de bases de datos. Un ejemplo es Recuperar los nombres de empleados que trabajan en **todos** los proyectos en los que trabaja 'John Smith'. Para expresar esta consulta usando la operación DIVISION , proceda de la siguiente manera. Primero, recupere el

lista de números de proyecto en los que trabaja 'John Smith' en la relación intermedia SMITH_PNOS :

```
SMITH ← σ Fname = 'John' ∧ Lname = 'Smith' ( EMPLEADO )
SMITH_PNOS ← π Pno ( WORKS_ON Essn = Ssn SMITH )
```

A continuación, cree una relación que incluya una tupla $\langle Pno, Essn \rangle$ siempre que el empleado cuyo Ssn es Essn trabaja en el proyecto cuyo número es Pno en el intermedio relación SSN_PNOS :

```
SSN_PNOS ← π Essn, Pno ( TRABAJA_ON )
```

Finalmente, aplique la operación DIVISION a las dos relaciones, lo que da la deseada números de seguro social de los empleados:

```
SSNS ( Ssn ) ← SSN_PNOS ÷ SMITH_PNOS
RESULTADO ← π Fname, Lname ( SSNS * EMPLOYEE )
```

Las operaciones anteriores se muestran en la Figura 8.8 (a).

En general, la operación DIVISION se aplica a dos relaciones $R(Z) \div S(X)$, donde el los atributos de S son un subconjunto de los atributos de R; es decir, $X \subseteq Z$. Sea Y el conjunto de atributos de R que no son atributos de S; es decir, $Y = Z - X$ (y por tanto $Z = X \cup Y$).

Figura 8.8
La operación DIVISION. (a) Dividiendo SSN_PNOS por SMITH_PNOS. (b) $T \leftarrow R \div S$.

(un)			(segundo)		
SSN_PNOS		SMITH_PNOS	R	S	
Essn	Pno	Pno	UN	segundo	UN
123456789	1	1	a1	b1	a1
123456789	2	2	a2	b1	a2
666884444	3		a3	b1	a3
453453453	1		a4	b1	
SSNS			T		

453453453	2		a1	b2	
333445555	2	Ssn	a3	b2	segundo
333445555	3	123456789	a2	b3	b1
333445555	10	453453453	a3	b3	b4
333445555	20		a4	b3	
999887777	30		a1	b4	
999887777	10		a2	b4	
987987987	10		a3	b4	
987987987	30				
987654321	30				
987654321	20				
888665555	20				

8.3 Operaciones relacionales binarias: JOIN y DIVISION 257

El resultado de DIVISION es una relación $T(Y)$ que incluye una tupla t si las tuplas t_R aparecen en R con $t_R[Y] = t$, y con $t_R[X] = t_S$ para cada tupla t_S en S . Esto significa que, para un tupla t para aparecer en el resultado T de la DIVISIÓN, los valores en t deben aparecer en R en combinación con cada tupla en S . Tenga en cuenta que en la formulación de la DIVISIÓN operación, las tuplas en la relación del denominador S restringen la relación del numerador R seleccionando aquellas tuplas en el resultado que coinciden con todos los valores presentes en el denominador. No es necesario saber cuáles son esos valores, ya que pueden ser calculado por otra operación, como se ilustra en la relación SMITH_PNOS en el ejemplo anterior.

La figura 8.8 (b) ilustra una operación de DIVISIÓN donde $X = \{A\}$, $Y = \{B\}$ y $Z = \{A, B\}$. Observe que las tuplas (valores) b_1 y b_4 aparecen en R en combinación con los tres tuplas en S ; por eso aparecen en la relación resultante T . Todos los demás valores de B en R no aparecen con todas las tuplas en S y no están seleccionadas: b_2 no aparece con un 2 y b_3 no aparece con un 1 .

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

$$\begin{aligned} T1 &\leftarrow \pi_Y(R) \\ T2 &\leftarrow \pi_Y((S \times T1) - R) \\ T &\leftarrow T1 - T2 \end{aligned}$$

La operación DIVISION se define por conveniencia para tratar consultas que implican la cuantificación universal (consulte la Sección 8.6.7) o la condición total. Más Las implementaciones de RDBMS con SQL como lenguaje de consulta principal no implementar la división. SQL tiene una forma indirecta de tratar el tipo de consulta recién ilustrado (consulte la Sección 7.1.4, consultas Q3A y Q3B). La tabla 8.1 enumera los diversos operaciones básicas de álgebra relacional que hemos discutido.

8.3.5 Notación para árboles de consultas

En esta sección describimos una notación que se usa típicamente en DBMS relacionales (RDBMS) para representar consultas internamente. La notación se llama árbol de consulta o, a veces, se conocido como árbol de evaluación de consultas o árbol de ejecución de consultas. Incluye lo relacional operaciones de álgebra que se están ejecutando y se utiliza como una posible estructura de datos para el representación interna de la consulta en un RDBMS.

Un **árbol de consulta** es una estructura de datos de árbol que corresponde a una expresión de álgebra relacional. Representa las relaciones de entrada de la consulta como nodos hoja del árbol, y representa Resiente las operaciones del álgebra relacional como nodos internos. Una ejecución del El árbol de consultas consiste en ejecutar una operación de nodo interno siempre que sus operandos (representados por sus nodos secundarios) están disponibles, y luego reemplazando ese nodo interno por la relación que resulta de ejecutar la operación. La ejecución termina cuando se ejecuta el nodo raíz y produce la relación de resultado para la consulta.

La Figura 8.9 muestra un árbol de consultas para la Consulta 2 (consulte la Sección 6.3.1): Para cada proyecto ubicado en 'Stafford', enumere el número de proyecto, el número de departamento de control y el apellido, la dirección y la fecha de nacimiento del gerente de departamento. Esta consulta está especificada

Tabla 8.1 Operaciones del álgebra relacional

OPERACIÓN	PROPÓSITO	NOTACIÓN
SELECCIONE	Selecciona todas las tuplas que satisfacen la selección. condición de una relación R.	σ < condición de selección > (R)
PROYECTO	Produce una nueva relación con sólo algunos de los atributos de R y elimina las tuplas duplicadas.	π < lista de atributos > (R)
THETA ÚNETE	Produce todas las combinaciones de tuplas de R ₁ y R ₂ que satisfacen la condición de unión.	R_1 < condición de unión > R ₂
EQUIJOIN	Produce todas las combinaciones de tuplas de R ₁ y R ₂ que satisfacen una condición de unión con solo comparaciones de igualdad.	R_1 < condición de unión > R ₂ , OR R_1 (< atributos de unión 1 >), (< atributos de unión 2 >) R ₂
UNIÓN NATURAL	Igual que EQUIJOIN excepto que los atributos de unión de R ₂ no se incluyen en la relación resultante; si los atributos de combinación tienen los mismos nombres, no es necesario especificarlo en absoluto.	$R_1 * < \text{condición de unión} > R_2$, $O R_1 * (< \text{atributos de combinación 1} >), (< \text{atributos de unión 2} >)$ $R_2 \circ R_1 * R_2$
UNIÓN	Produce una relación que incluye todas las tuplas. en R ₁ o R ₂ o tanto R ₁ como R ₂ ; R ₁ y R ₂ deben ser compatible con la unión.	$R_1 \cup R_2$
INTERSECCIÓN	Produce una relación que incluye todas las tuplas. tanto en R ₁ como en R ₂ ; R ₁ y R ₂ deben estar unidos compatible.	$R_1 \cap R_2$

DIFERENCIA	Produce una relación que incluye todas las tuplas. en R_1 que no están en R_2 ; R_1 y R_2 deben ser Unión compatible.	$R_1 - R_2$
PRODUCTO CARTESIANO	Produce una relación que tiene los atributos de R_1 y R_2 e incluye como tuplas todas las posibles combinaciones de tuplas de R_1 y R_2 .	$R_1 \times R_2$
DIVISIÓN	Produce una relación $R(X)$ que incluye todas las tuplas. t $[X]$ en $R_1(Z)$ que aparecen en R_1 en combinación con cada tupla de $R_2(Y)$, donde $Z = X \cup Y$.	$R_1(Z) \div R_2(Y)$

en el esquema relacional de la Figura 5.5 y corresponde al siguiente esquema relacional expresión de álgebra:

$\pi_{Pnumber, Dnum, Lname, Address, Bdate}(((\sigma_{Ubicación = 'Stafford'}(PROYECTO)))$
 $Dnum = Dnumber(DEPARTAMENTO)) \text{ Mgr_ssn = Ssn(EMPLEADO)})$

En la figura 8.9, los tres nodos hoja P , D y E representan las tres relaciones PROYECTO , DEPARTAMENTO y EMPLEADO . Las operaciones de álgebra relacional en la expresión son representado por los nodos internos del árbol. El árbol de consulta significa un orden explícito de ejecución. ción en el siguiente sentido. Para ejecutar Q2 , el nodo marcado (1) en la Figura 8.9 debe comenzar la ejecución antes del nodo (2) porque algunas tuplas resultantes de operaciones La opción (1) debe estar disponible antes de que podamos comenzar a ejecutar la operación (2). Similar,

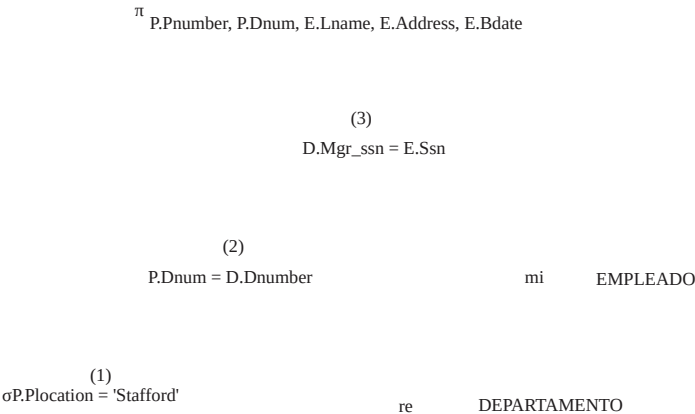


Figura 8.9
Árbol de consultas correspondiente al álgebra relacional expresión para Q2.

el nodo (2) debe comenzar a ejecutarse y producir resultados antes de que el nodo (3) pueda iniciar la ejecución, y así. En general, un árbol de consultas ofrece una buena representación visual y comprensión de la consulta en términos de las operaciones relacionales que utiliza y se recomienda como un medio adicional para expresar consultas en álgebra relacional. Revisaremos la consulta árboles cuando analizamos el procesamiento y la optimización de consultas en los capítulos 18 y 19.

8.4 Operaciones relacionales adicionales

Algunas solicitudes de bases de datos comunes, que se necesitan en aplicaciones comerciales para RDBMS: no se puede realizar con las operaciones originales de álgebra relacional descrito en las Secciones 8.1 a 8.3. En esta sección definimos operaciones adicionales para expresar estas solicitudes. Estas operaciones realzan el poder expresivo de el álgebra relacional original.

8.4.1 Proyección generalizada

La operación de proyección generalizada extiende la operación de proyección al permitir funciones de los atributos que se incluirán en la lista de proyección. La forma generalizada se puede expresar como:

$$\pi_{F_1, F_2, \dots, F_n}(R)$$

donde F_1, F_2, \dots, F_n son funciones sobre los atributos en relación R y pueden involucrar operaciones aritméticas y valores constantes. Esta operación es útil cuando se desarrolla informes de operaciones donde los valores calculados deben producirse en las columnas de un resultado de la consulta.

Como ejemplo, considere la relación

EMPLEADO (Ssn, Salario, Deducción, Años_servicio)

Es posible que se requiera un informe para mostrar

Salario neto = Salario - Deducción ,

Bonificación = 2000 * años_servicio , y

Impuesto = 0,25 * Salario

Entonces, una proyección generalizada combinada con un cambio de nombre se puede usar de la siguiente manera:

INFORME \leftarrow ρ (Ssn, Net_salary, Bonus, Tax) (π Ssn, Salario - Deducción , 2000 * Years_service ,

0.25 * Salario (EMPLEADO))

8.4.2 Funciones agregadas y agrupación

Otro tipo de solicitud que no se puede expresar en el álgebra relacional básica es para especificar **funciones matemáticas agregadas** en colecciones de valores de la base de datos. Ejemplos de tales funciones incluyen recuperar el salario promedio o total de todos los empleados o el número total de tuplas de empleados. Estas funciones se utilizan en consultas estadísticas sencillas que resumen la información de la base de datos tuplas. Las funciones comunes aplicadas a colecciones de valores numéricos incluyen SUM , PROMEDIO , MÁXIMO y MÍNIMO . La función COUNT se utiliza para contar tuplas o valores.

Otro tipo común de solicitud implica agrupar las tuplas en una relación por el valor de algunos de sus atributos y luego aplicar una función agregada independiente con atención a cada grupo. Un ejemplo sería agrupar tuplas EMPLEADOS por Dno , por lo que que cada grupo incluye las tuplas para los empleados que trabajan en el mismo departamento. Luego podemos enumerar cada valor de Dno junto con, digamos, el salario promedio de los empleados dentro del departamento, o el número de empleados que trabajan en el departamento.

Podemos definir una operación de FUNCIÓN AGREGADA , usando el símbolo \bowtie (pro-script sustantivo F) ⁷ , para especificar este tipo de solicitudes de la siguiente manera:

$$\langle \text{atributos de agrupación} \rangle \bowtie \langle \text{lista de funciones} \rangle (R)$$

donde $\langle \text{atributos de agrupación} \rangle$ es una lista de atributos de la relación especificada en R, y $\langle \text{lista de funciones} \rangle$ es una lista de pares ($\langle \text{función} \rangle \langle \text{atributo} \rangle$). En cada uno de esos pares, $\langle \text{función} \rangle$ es una de las funciones permitidas, como SUMA , PROMEDIO , MÁXIMO , MÍNIMO , COUNT —y $\langle \text{attribute} \rangle$ es un atributo de la relación especificada por R. El resultado La relación tiene los atributos de agrupación más un atributo para cada elemento de la función. lista. Por ejemplo, para recuperar cada número de departamento, el número de empleados en el departamento, y su salario promedio, mientras que cambia el nombre de los atributos resultantes como indicado a continuación, escribimos:

$$\rho \ R \ (\ Dno, No_of_employees, Promedio_sal \) \ (\ Dno \ \bowtie \ COUNT \ Ssn, PROMEDIO \ Salario \ (\ EMPLEADO \))$$

⁷ No existe una notación única acordada para especificar funciones agregadas. En algunos casos, un "guión A" se utiliza.

R			(segundo)		
(un)	Dno	Cantidad de Empleados	Dno	Count_ssn	Salario promedio
	5	4	5	4	33250
	4	3	4	3	31000
	1	1	1	1	55000
(C)	Count_ssn	Salario promedio			
	8	35125			

Figura 8.10

La operación de función agregada.

a. ρ R (Dno, No_of_employees, Average_sal) (Dno \bowtie COUNT Ssn, PROMEDIO Salario (EMPLEADO)).

segundo. Dno \bowtie COUNT Ssn, Salario PROMEDIO (EMPLEADO).

C. \bowtie COUNT Ssn, Salario PROMEDIO (EMPLEADO).

El resultado de esta operación en la relación EMPLEADO de la Figura 5.6 se muestra en Figura 8.10 (a).

En el ejemplo anterior, especificamos una lista de nombres de atributos, entre paréntesis, en la operación RENAME, para la relación resultante R. Si no se aplica ningún cambio de nombre, entonces los atributos de la relación resultante que corresponden a la lista de funciones serán cada uno será la concatenación del nombre de la función con el nombre del atributo en la forma <función> _ <atributo>. ⁸ Por ejemplo, la figura 8.10 (b) muestra el resultado de la siguiente operación de bajada:

Dno \bowtie COUNT Ssn, Salario PROMEDIO (EMPLEADO)

Si no se especifican atributos de agrupación, las funciones se aplican a todas las tuplas en la relación, por lo que la relación resultante tiene una única tupla. Por ejemplo, Figure 8.10 (c) muestra el resultado de la siguiente operación:

\bowtie COUNT Ssn , Salario PROMEDIO (EMPLEADO)

Es importante señalar que, en general, los duplicados no se eliminan cuando un se aplica la función agregada; De esta manera, la interpretación normal de funciones tales como se calcula SUMA y PROMEDIO. ⁹ Sin embargo, los valores NULL no se consideran en la agregación, como discutimos en la Sección 7.1.7. Vale la pena enfatizar que el El resultado de aplicar una función agregada es una relación, no un número escalar, incluso si tiene un solo valor. Esto hace que el álgebra relacional sea una matemática cerrada. sistema.

⁸ Tenga en cuenta que esta es una notación arbitraria, consistente con lo que haría SQL.

⁹ En SQL, la opción de eliminar duplicados antes de aplicar la función agregada está disponible incluida la palabra clave DISTINCT (consulte la Sección 4.4.4).

8.4.3 Operaciones de cierre recursivo

Otro tipo de operación que, por lo general, no se puede especificar en el original básico el álgebra relacional es **cierre recursivo**. Esta operación se aplica a un **recursivo relación** entre tuplas del mismo tipo, como la relación entre un

empleado y supervisor. Esta relación está descrita por la clave externa Super_ssn de la relación EMPLEADO en las Figuras 5.5 y 5.6, y relaciona cada tupla de empleado (en el papel de supervisado) a otra tupla de empleado (en el papel de supervisor). Un ejemplo de una operación recursiva es recuperar todos los supervisados de un empleado e en todos los niveles, es decir, todos los empleados e ' supervisados directamente por e, todos los empleados ees e' supervisados directamente por cada empleado e ', todos los empleados e " ' supervisan directamente visualizado por cada empleado e ", y así sucesivamente.

Es relativamente sencillo en el álgebra relacional especificar todos los empleados supervisado por e en un nivel específico al unirse a la mesa consigo mismo uno o más veces. Sin embargo, es difícil especificar todos los supervisados en todos los niveles. Por ejemplo, para especificar los Ssn de todos los empleados e ' supervisados directamente, en el nivel uno, por el empleado e cuyo nombre es 'James Borg' (ver Figura 5.6), podemos aplicar lo siguiente: operación de ing:

```
BORG_SSN ← π Ssn (σ Fname = 'James' Y Lname = 'Borg' ( EMPLEADO ))
SUPERVISIÓN ( Ssn1 , Ssn2 ) ← π Ssn , Super_ssn ( EMPLEADO )
RESULT1 ( Ssn ) ← π Ssn1 ( SUPERVISIÓN Ssn2 = Ssn BORG_SSN)
```

Para recuperar a todos los empleados supervisados por Borg en el nivel 2, es decir, todos los empleados e " supervisado por algún empleado e ' que es supervisado directamente por Borg; podemos solicitar otro JOIN al resultado de la primera consulta, como sigue:

```
RESULTADO2 ( Ssn ) ← π Ssn1 ( SUPERVISIÓN Ssn2 = Ssn RESULT1 )
```

Para que ambos grupos de empleados sean supervisados en los niveles 1 y 2 por 'James Borg', podemos aplique la operación UNION a los dos resultados, de la siguiente manera:

```
RESULTADO ← RESULT2 U RESULT1
```

Los resultados de estas consultas se ilustran en la Figura 8.11. Aunque es posible recuperar empleados en cada nivel y luego tomar su UNIÓN, no podemos, en general, especificar una consulta como "recuperar los supervisados de 'James Borg' en todos los niveles" sin utilizando un mecanismo de bucle a menos que sepamos el número máximo de niveles.¹⁰ Se ha propuesto una operación denominada cierre transitivo de relaciones para Pute la relación recursiva en la medida en que prosiga la recursividad.

8.4.4 Operaciones OUTER JOIN

A continuación, discutimos algunas extensiones adicionales a la operación JOIN que son necesarias. Es necesario especificar ciertos tipos de consultas. Las operaciones JOIN descritas anteriormente coincide con las tuplas que satisfacen la condición de combinación. Por ejemplo, para un NATURAL JOIN

¹⁰ El estándar SQL3 incluye sintaxis para cierre recursivo.

SUPERVISIÓN	
(El Ssn de Borg es 888665555)	
(Ssn)	(Super_ssn)
Ssn1	Ssn2
123456789	333445555
333445555	888665555
999887777	987654321
987654321	888665555
666884444	333445555
453453453	333445555
987987987	987654321
888665555	nulo

RESULTADO1	RESULTADO2	RESULTADO
Ssn	Ssn	Ssn
333445555	123456789	123456789
987654321	999887777	999887777
(Supervisado por Borg)	666884444	666884444
	453453453	453453453
	987987987	987987987
	(Supervisado por Subordinados de Borg)	333445555
		987654321
(RESULT1 U RESULT2)		

Figura 8.11
Un recursivo de dos niveles consulta.

operación $R * S$, solo tuplas de R que tienen tuplas coincidentes en S, y viceversa versa — aparece en el resultado. Por lo tanto, tuplas sin una tupla coincidente (o relacionada) se eliminan del resultado de JOIN . Tuplas con valores NULL en el atributo de combinación también se eliminan los butes. Este tipo de combinación, donde las tuplas sin coincidencia se eliminan inated, se conoce como **combinación interna** . Las operaciones de unión que describimos anteriormente en La sección 8.3 son todas combinaciones internas. Esto equivale a la pérdida de información si el usuario quiere que el resultado de JOIN incluya todas las tuplas en una o más de las combinaciones relaciones con los ponentes.

Se desarrolló un conjunto de operaciones, llamadas **combinaciones externas** , para el caso en el que el usuario quiere mantener todas las tuplas en R, o todas las de S, o todas las de ambas relaciones en el resultado de JOIN , independientemente de si tienen o no tuplas coincidentes en la otra relación. Esto satisface la necesidad de consultas en las que tuplas de dos Las tablas deben combinarse haciendo coincidir las filas correspondientes, pero sin perder cualquier tupla por falta de valores coincidentes. Por ejemplo, supongamos que queremos una lista de todos los nombres de los empleados, así como el nombre de los departamentos que administran si pasar a administrar un departamento; si no gestionan uno se lo podemos indicar

Figura 8.12

El resultado de una IZQUIERDA
Operación OUTER JOIN.

RESULTADO

Fname	Minit	Lname	Dname
Juan	segundo	Herrero	NULO
Franklin	T	Wong	Investigación
Alicia	J	Zelaya	NULO
Jennifer	S	Wallace	Administración
Ramesh	K	Narayan	NULO
Joyce	UN	Inglés	NULO
Ahmad	V	Jabbar	NULO
James	mi	Borg	Sede

con un valor NULO . Podemos aplicar una operación LEFT OUTER JOIN , denotada por, a recuperar el resultado de la siguiente manera:

$$\begin{aligned} \text{TEMP} &\leftarrow (\text{EMPLEADO} \quad \text{Ssn} = \text{Mgr_ssn} \text{ DEPARTAMENTO}) \\ \text{RESULTADO} &\leftarrow \pi_{\text{Fname}, \text{Minit}, \text{Lname}, \text{Dname}} (\text{TEMP}) \end{aligned}$$

La operación LEFT OUTER JOIN mantiene cada tupla en la primera relación R, o izquierda, en RS; si no se encuentra una tupla coincidente en S, entonces los atributos de S en el resultado de la combinación se rellenan o relleno con valores NULL . El resultado de estas operaciones se muestra en la Figura 8.12.

Una operación similar, RIGHT OUTER JOIN , denotado por, mantiene cada tupla en la segunda, o derecha, relación S en el resultado de R S. Una tercera operación, FULL OUTER JOIN , denotado por, mantiene todas las tuplas en las relaciones izquierda y derecha cuando no se encuentran tuplas coincidentes, relleniéndolas con valores NULL según sea necesario. El tres Las operaciones de combinación externa son parte del estándar SQL2 (consulte la Sección 7.1.6). Estos operarios más tarde se proporcionaron las descripciones como una extensión del álgebra relacional en respuesta a la necesidad típica en aplicaciones comerciales de mostrar información relacionada de múltiples tablas de forma exhaustiva. A veces, un informe completo de datos de varias tablas es requerido si hay valores coincidentes o no.

8.4.5 La operación UNIÓN EXTERIOR

La operación OUTER UNION fue desarrollada para tomar la unión de tuplas de dos relaciones que tienen algunos atributos comunes, pero no son compatibles con la unión (tipo). Esta operación tomará la UNIÓN de tuplas en dos relaciones R (X, Y) y S (X, Z) que son **parcialmente compatibles** , lo que significa que solo algunos de sus atributos, digamos X, son Unión compatible. Los atributos que son compatibles con la unión se representan solo una vez en el resultado, y aquellos atributos que no son compatibles con la unión de La relación también se mantiene en la relación de resultado T (X, Y, Z). Por tanto, es lo mismo que un FULL OUTER JOIN en los atributos comunes.

Se dice que dos tuplas t_1 en R y t_2 en S **coinciden** si $t_1[X] = t_2[X]$. Estos serán combined (unioned) en una sola tupla en t. Tuplas en cualquier relación que no tienen la tupla coincidente en la otra relación se rellena con valores NULL . Por ejemplo, un

UNIÓN EXTERIOR se puede aplicar a dos relaciones cuyos esquemas son ESTUDIANTE (Nombre , Ssn , Departamento , Asesor) e INSTRUCTOR (Nombre , Ssn , Departamento , Rango). Tuplas de las dos relaciones se emparejan basándose en tener la misma combinación de valores de los atributos compartidos: Nombre , Ssn , Departamento . La relación resultante, STUDENT_OR_INSTRUCTOR , tendrá los siguientes atributos:

STUDENT_OR_INSTRUCTOR (Nombre , Ssn , Departamento , Asesor , Rango)

Todas las tuplas de ambas relaciones se incluyen en el resultado, pero las tuplas con el La misma combinación (Nombre , Ssn , Departamento) aparecerá solo una vez en el resultado. Las tuplas que aparecen solo en ESTUDIANTE tendrán un NULO para el atributo Rango , mientras que las tuplas que aparecen solo en INSTRUCTOR tendrán un NULL para el atributo Asesor . Una tupla que existe en ambas relaciones, que representa a un estudiante que también es un profesor, tendrá valores para todos sus atributos. ¹¹

Tenga en cuenta que la misma persona aún puede aparecer dos veces en el resultado. Por ejemplo, nosotros podríamos tener un estudiante graduado en el departamento de Matemáticas que sea profesor en el departamento de Ciencias de la Computación. Aunque las dos tuplas que representan esa persona en ESTUDIANTE e INSTRUCTOR tendrán los mismos valores de (Nombre , Ssn), no estarán de acuerdo con el valor del Departamento , por lo que no se igualarán. Esto es porque Departamento tiene dos significados diferentes en ESTUDIANTE (el departamento donde el estudiante estudia) e INSTRUCTOR (el departamento donde la persona trabaja como profesor). Si quisiéramos aplicar la UNIÓN EXTERIOR en base al mismo (Nombre , Ssn) combinación solamente, debemos cambiar el nombre del atributo Departamento en cada tabla para reflejar que tienen diferentes significados y los designamos como no parte de los atributos compatibles con la unión. Por ejemplo, podríamos cambiar el nombre de los atributos como Mayor Departamento en ESTUDIANTE y Departamento de Trabajo en INSTRUCTOR .

8.5 Ejemplos de consultas en álgebra relacional

Los siguientes son ejemplos adicionales para ilustrar el uso de las operaciones de álgebra relacional. Todos los ejemplos se refieren a la base de datos de la Figura 5.6. En general, la misma consulta se puede formular de diversas formas utilizando las distintas operaciones. Lo haremos enunciar cada consulta de una manera y deje que el lector cree un equivalente formulaciones.

Consulta 1. Obtenga el nombre y la dirección de todos los empleados que trabajan para 'Departamento de Investigación'.

```
RESEARCH_DEPT ← σ Dname = 'Investigación' ( DEPARTAMENTO )
RESEARCH_EMPS ← ( RESEARCH_DEPT Dnumber = Dno EMPLOYEE )
RESULTADO ← π Fname , Lname , Address ( RESEARCH_EMPS )
```

Como una única expresión en línea, esta consulta se convierte en:

```
π Fname , Lname , Address ( σ Dname = 'Research' ( DEPARTMENT Dnumber = Dno ( EMPLOYEE ) )
```

¹¹ Tenga en cuenta que OUTER UNION es equivalente a FULL OUTER JOIN si los atributos de combinación son todos los atributos de las dos relaciones.

Esta consulta se podría especificar de otras formas; por ejemplo, el orden de JOIN y las operaciones SELECT podrían revertirse, o el JOIN podría ser reemplazado por un

NATURAL JOIN después de cambiar el nombre de uno de los atributos de combinación para que coincida con la otra combinación Nombre del Atributo.

Consulta 2. Para cada proyecto ubicado en 'Stafford', enumere el número de proyecto, el número del departamento de pesca y el apellido, dirección, y fecha de nacimiento.

```
STAFFORD_PROJS ← σ Ubicación = 'Stafford' ( PROYECTO )
CONTR_DEPTS ← ( STAFFORD_PROJS Dnum = Dnumber DEPARTMENT )
PROJ_DEPT_MGRS ← ( CONTR_DEPTS Mgr_ssn = SsnE MPLOYEE )
RESULTADO ← π Pnumber , Dnum , Lname , Address , Bdate ( PROJ_DEPT_MGRS )
```

En este ejemplo, primero seleccionamos los proyectos ubicados en Stafford, luego los unimos con sus departamentos de control, y luego unir el resultado con el departamento de administración. Finalmente, aplicamos una operación de proyecto sobre los atributos deseados.

Consulta 3. Busque los nombres de los empleados que trabajan en todos los proyectos controlados por departamento número 5.

```
DEPT5_PROJS ← ρ ( Pno ) ( π Pnumber ( σ Dnum = 5 ( PROYECTO ) ) )
EMP_PROJ ← ρ ( Ssn , Pno ) ( π Essn , Pno ( WORKS_ON ) )
RESULT_EMP_SSNS ← EMP_PROJ ÷ DEPT5_PROJS
RESULTADO ← π Lname , Fname ( RESULT_EMP_SSNS * EMPLOYEE )
```

En esta consulta, primero creamos una tabla DEPT5_PROJS que contiene el proyecto números de todos los proyectos controlados por el departamento 5. Luego creamos una tabla EMP_PROJ que contiene tuplas (Ssn, Pno) y aplica la operación de división. aviso que cambiamos el nombre de los atributos para que se usen correctamente en la división operación. Por último, nos sumamos el resultado de la división, que es válida sólo Ssn Values, con la tabla EMPLOYEE para recuperar los atributos Fname, Lname de EMPLEADO.

Consulta 4. Haga una lista de números de proyecto para proyectos que involucran a un empleado cuyo apellido es 'Smith', ya sea como trabajador o como gerente del departamento que controla el proyecto.

```
SMITHS ( Essn ) ← π Ssn ( σ Lname = 'Smith' ( EMPLEADO ) )
SMITH_WORKER_PROJS ← π Pno ( WORKS_ON * SMITHS )
MGRS ← π Lname , Dnumber ( EMPLEADO Ssn = Mgr_ssn DEPARTAMENTO )
SMITH_MANAGED_DEPTS ( Dnum ) ← π Dnumber ( σ Lname = 'Smith' ( MGRS ) )
SMITH_MGR_PROJS ( Pno ) ← π Pnumber ( SMITH_MANAGED_DEPTS * PROJECT )
RESULTADO ← ( SMITH_WORKER_PROJS U SMITH_MGR_PROJS )
```

En esta consulta, recuperamos los números de proyecto para proyectos que involucran a un empleado nombró a Smith como trabajador en SMITH_WORKER_PROJS . Luego recuperamos el proyecto números ect para proyectos que involucran a un empleado llamado Smith como gerente de la departamento que controla el proyecto en SMITH_MGR_PROJS . Finalmente, aplicamos el

Operación UNION en SMITH_WORKER_PROJS y SMITH_MGR_PROJS . Como soltero expresión en línea, esta consulta se convierte en:

$$\pi_{Pno} (\text{WORKS_ON} \text{ Essn} = \text{Ssn} (\pi_{\text{Ssn}} (\sigma_{\text{Lname} = \text{'Smith'}} (\text{EMPLEADO})))) \cup \pi_{Pno} ((\pi_{\text{Dnumber}} (\sigma_{\text{Lname} = \text{'Smith'}} (\pi_{\text{Lname}, \text{Dnumber}} (\text{EMPLOYEE}))))$$

$$\text{Ssn} = \text{Mgr_ssn DEPARTAMENTO})) \text{ Dnum-ber} = \text{Dnum PROYECTO})$$

Consulta 5. Enumere los nombres de todos los empleados con dos o más dependientes.

Estrictamente hablando, esta consulta no se puede realizar en el relacional básico (original) álgebra. Tenemos que usar la operación FUNCIÓN AGREGADA con la CUENTA función agregada. Suponemos que los dependientes del mismo empleado tienen valores distintos de Dependent_name .

$$T1 (\text{Ssn}, \text{No_of_dependientes}) \leftarrow \text{Essn} \Join_{\text{COUNT nombre_dependiente} (\text{DEPENDIENTE})}$$

$$T2 \leftarrow \sigma_{\text{No_de_dependientes} > 2} (T1)$$

$$\text{RESULTADO} \leftarrow \pi_{\text{Lname}, \text{Fname}} (T2 * \text{EMPLEADO})$$

Consulta 6. Recupere los nombres de los empleados que no tienen dependientes.

Este es un ejemplo del tipo de consulta que usa MINUS (SET DIFFERENCE) operación.

$$\text{ALL_EMPS} \leftarrow \pi_{\text{Ssn}} (\text{EMPLEADO})$$

$$\text{EMPS_WITH_DEPS} (\text{Ssn}) \leftarrow \pi_{\text{Essn}} (\text{DEPENDIENTE})$$

$$\text{EMPS_WITHOUT_DEPS} \leftarrow (\text{ALL_EMPS} - \text{EMPS_WITH_DEPS})$$

$$\text{RESULTADO} \leftarrow \pi_{\text{Lname}, \text{Fname}} (\text{EMPS_WITHOUT_DEPS} * \text{EMPLOYEE})$$

Primero recuperamos una relación con todos los Ssn de los empleados en ALL_EMPS . Entonces creamos una tabla con los Ssn de los empleados que tienen al menos un dependiente en

EMPS_WITH_DEPS . Luego aplicamos la operación SET DIFFERENCE para recuperar empleados Ssn s sin dependientes en EMPS_WITHOUT_DEPS , y finalmente unirse esto con EMPLOYEE para recuperar los atributos deseados. Como una única expresión en línea sion, esta consulta se convierte en:

$$\pi_{\text{Lname}, \text{Fname}} ((\pi_{\text{Ssn}} (\text{EMPLEADO}) - \rho_{\text{Ssn}} (\pi_{\text{Essn}} (\text{DEPENDIENTE}))) * \text{EMPLOYEE})$$

Consulta 7. Enumere los nombres de los gerentes que tienen al menos un dependiente.

$$\text{MGRS} (\text{Ssn}) \leftarrow \pi_{\text{Mgr_ssn}} (\text{DEPARTAMENTO})$$

$$\text{EMPS_WITH_DEPS} (\text{Ssn}) \leftarrow \pi_{\text{Essn}} (\text{DEPENDIENTE})$$

$$\text{MGRS_WITH_DEPS} \leftarrow (\text{MGRS} \cap \text{EMPS_WITH_DEPS})$$

$$\text{RESULTADO} \leftarrow \pi_{\text{Lname}, \text{Fname}} (\text{MGRS_WITH_DEPS} * \text{EMPLOYEE})$$

En esta consulta, recuperamos las Ssn de gerentes en MGRS y las Ssn de empleo.

ees con al menos un dependiente en EMPS_WITH_DEPS , luego aplicamos el SET

Operación de INTERSECCIÓN para obtener las Ssn de los gerentes que tienen al menos una dependiente.

Como mencionamos anteriormente, la misma consulta se puede especificar de muchas formas diferentes en álgebra relacional. En particular, las operaciones a menudo se pueden aplicar en varios pedidos. Además, algunas operaciones se pueden utilizar para reemplazar otras; por ejemplo, el

La operación de INTERSECCIÓN en Q7 se puede reemplazar por una JUNTA NATURAL . Como ejercicio intente realizar cada una de estas consultas de muestra utilizando diferentes operaciones. ¹² Mostramos cómo escribir consultas como expresiones únicas de álgebra relacional para consultas Q1 , Q4 y Q6 . Tratar para escribir las consultas restantes como expresiones únicas. En los capítulos 6 y 7 y en la sección En las secciones 8.6 y 8.7, mostramos cómo estas consultas están escritas en otros lenguajes relacionales.

8.6 El cálculo relacional de tuplas

En esta sección y en la siguiente, presentamos otro lenguaje de consulta formal para modelo relacional llamado **cálculo relacional** . Esta sección presenta el idioma conocido como **cálculo relacional de tuplas** , y la Sección 8.7 introduce una variación llamada **cálculo relacional de dominio** . En ambas variaciones del cálculo relacional, escribimos una expresión **declarativa** para especificar una solicitud de recuperación; por lo tanto, no hay descripción de cómo, o en qué orden, evaluar una consulta. Una expresión de cálculo especifica qué es ser recuperado en lugar de cómo recuperarlo. Por tanto, el cálculo relacional es considerado un lenguaje no **procesal** . Esto difiere del álgebra relacional, donde debemos escribir una secuencia de operaciones para especificar una solicitud de recuperación en un par-orden particular de aplicación de las operaciones; por tanto, puede considerarse como un **procedimiento** forma de formular una consulta. Es posible anidar operaciones de álgebra para formar un solo expresión; Sin embargo, siempre se especifica explícitamente un cierto orden entre las operaciones. ified en una expresión de álgebra relacional. Este orden también influye en la estrategia para evaluar la consulta. Una expresión de cálculo puede escribirse de diferentes formas, pero la forma en que está escrito no influye en cómo se debe evaluar una consulta.

Se ha demostrado que cualquier recuperación que pueda especificarse en la álgebra relacional básica bra también se puede especificar en cálculo relacional y viceversa; en otras palabras, el **El poder expresivo** de las lenguas es idéntico. Esto llevó a la definición de la con- excepto de un lenguaje relacionamente completo. Se considera un lenguaje de consulta relacional L **relacionalmente completo** si podemos expresar en L cualquier consulta que pueda expresarse en cálculo relacional. La integridad relacional se ha convertido en una base importante para comparando el poder expresivo de los lenguajes de consulta de alto nivel. Sin embargo, como vimos En la Sección 8.4, ciertas consultas requeridas con frecuencia en aplicaciones de bases de datos expresarse en álgebra relacional básica o cálculo. La mayoría de los lenguajes de consulta relacionales son relacionamente completos pero tienen más poder expresivo que el álgebra relacional o cálculo relacional debido a operaciones adicionales como funciones agregadas, agrupación y ordenación. Como mencionamos en la introducción de este capítulo, el El cálculo relacional es importante por dos razones. Primero, tiene una base firme en matemática lógica matemática. En segundo lugar, el lenguaje de consulta estándar (SQL) para RDBMS tiene su fundamento en el cálculo relacional de tuplas.

Nuestros ejemplos se refieren a la base de datos que se muestra en las Figuras 5.6 y 5.7. Usaremos el mismas consultas que se utilizaron en la Sección 8.5. Las secciones 8.6.6, 8.6.7 y 8.6.8 discuten tratar con cuantificadores universales y cuestiones de seguridad de expresión. Estudiantes entre citado en una introducción básica al cálculo relacional de tuplas puede omitir estas secciones.

¹² Cuando se optimizan las consultas (véanse los Capítulos 18 y 19), el sistema elegirá una secuencia particular de operaciones que corresponde a una estrategia de ejecución que se puede ejecutar de manera eficiente.

8.6.1 Variables de tupla y relaciones de rango

El cálculo relacional de tuplas se basa en especificar una serie de **variables de tuplas**.

Cada variable de tupla generalmente varía sobre una relación de base de datos particular, lo que significa que la variable puede tomar como valor cualquier tupla individual de esa relación. Un simple La consulta de cálculo relacional de tupla tiene la forma:

$$\{t \mid \text{COND}(t)\}$$

donde t es una variable de tupla y $\text{COND}(t)$ es una expresión condicional (booleana)

t que se evalúa como VERDADERO o FALSO para diferentes asignaciones de tuplas a la variable t . El resultado de tal consulta es el conjunto de todas las tuplas t que evalúan como $\text{COND}(t)$ a TRUE. Se dice que estas tuplas **satisfacen** $\text{COND}(t)$. Por ejemplo, para encontrar todos los empleados cuyo salario es superior a \$ 50 000, podemos escribir el siguiente cálculo de tupla la expresión:

$$\{t \mid \text{EMPLEADO}(t) \text{ Y } t.\text{Salario} > 50000\}$$

La condición $\text{EMPLEADO}(t)$ especifica que la **relación de rango** de la variable tupla t es

EMPLEADO . Cada tupla de EMPLEADO t que satisface la condición $t.\text{Salario} > 50000$ voluntad ser recuperado. Note que t . Atributo de referencias de salario Salario de la variable tupla t ; esta La notación se asemeja a cómo los nombres de atributos se califican con nombres de relación o alias. en SQL, como vimos en el Capítulo 6. En la notación del Capítulo 5, t . El salario es el mismo que escribiendo $t[\text{Salario}]$.

La consulta anterior recupera todos los valores de atributo para cada tupla de EMPLEADO seleccionada t . Para recuperar solo algunos de los atributos, por ejemplo, el nombre y el apellido, escribimos

$$t.\text{Fname}, t.\text{Lname} \mid \text{EMPLEADO}(t) \text{ Y } t.\text{Salario} > 50000\}$$

De manera informal, necesitamos especificar la siguiente información en un cálculo relacional de tuplas la expresión:

- Para cada variable tupla t , la **relación de rango** R de t . Este valor está especificado por una condición de la forma $R(t)$. Si no especificamos una relación de rango, entonces la variable t abarcará todas las tuplas posibles "en el universo", ya que no es restringido a cualquier relación.
- Una condición para seleccionar combinaciones particulares de tuplas. Como variables de tupla rango sobre sus respectivas relaciones de rango, la condición se evalúa para cada combinación posible de tuplas para identificar las **combinaciones seleccionadas** para lo cual la condición se evalúa como VERDADERA.
- Un conjunto de atributos a recuperar, los **atributos solicitados**. Los valores de estos atributos se recuperan para cada combinación seleccionada de tuplas.

Antes de discutir la sintaxis formal del cálculo relacional de tuplas, considere otra consulta.

Consulta 0. Recupera la fecha de nacimiento y la dirección del empleado (o empleados) cuyo nombre es John B. Smith.

Q0: {t. Bdate, t. Direccion | EMPLEADO (t) Y t. Fname = 'John' Y t. Minit = 'B'
Y t. Lname = 'Smith'}

Página 32

270

Capítulo 8 El álgebra relacional y el cálculo relacional

En el cálculo relacional de tuplas, primero especificamos los atributos solicitados t. Bdate y t. Dirección para cada tupla seleccionada t. Luego especificamos la condición para seleccionar una tupla siguiendo la barra (|), es decir, que sea una tupla de la relación EMPLEADO cuya Los valores de los atributos Fname, Minit y Lname son 'John', 'B' y 'Smith', respectivamente.

8.6.2 Expresiones y fórmulas en cálculo relacional de tuplas

Una **expresión** general del cálculo relacional de tuplas tiene la forma

$$\{t_1.A_j, t_2.A_k, \dots, t_n.A_m \mid \text{COND}(t_1, t_2, \dots, t_n, t_{n+1}, t_{n+2}, \dots, t_{n+m})\}$$

donde $t_1, t_2, \dots, t_n, t_{n+1}, \dots, t_{n+m}$ son variables de tupla, cada A_i es un atributo de la relación en la que t_i varía, y COND es una **condición** o **fórmula** de la relación tupla cálculo relacional. Una fórmula se compone de **átomos** de cálculo de predicados, que pueden ser uno de los siguientes:

1. Un átomo de la forma $R(t_i)$, donde R es un nombre de relación y t_i es una variable tupla. poder. Este átomo identifica el rango de la variable tupla t_i como la relación cuyo nombre es R. Se evalúa como VERDADERO si t_i es una tupla en la relación R, y se evalúa como FALSO en caso contrario.
2. Un átomo de la forma $t_i.A \text{ op } t_j.B$, donde **op** es una de las operaciones de comparación en el conjunto $\{=, <, <=, >, \geq, \neq\}$, t_i y t_j son variables de tupla, A es un atributo de la relación en la que t_i varía, y B es un atributo de la relación en la que t_j varía.
3. Un átomo de la forma $t_i.A \text{ op } c \text{ o } c \text{ op } t_j.B$, donde **op** es uno de los operadores en el conjunto $\{=, <, <=, >, \geq, \neq\}$, t_i y t_j son variables de tupla, A es un atributo de la relación en la que t_i varía, B es un atributo de la relación en la que t_j varía, y c es un valor constante.

Cada uno de los átomos precedentes se evalúa como VERDADERO o FALSO para una combinación específica de tuplas; esto se llama el **valor de verdad** de un átomo. En general, una variable de tupla t abarca todas las tuplas posibles en el universo. Para átomos de la forma $R(t)$, si t es asignado a una tupla que es miembro de la relación R especificada, el átomo es VERDADERO; de lo contrario, es FALSO. En átomos de tipos 2 y 3, si se asignan las variables de tupla a tuplas de modo que los valores de los atributos especificados de las tuplas satisfagan las condiciones, entonces el átomo es VERDADERO.

Una **fórmula** (condición booleana) se compone de uno o más átomos conectados a través de los operadores lógicos \wedge , \vee y \neg y se define de forma recursiva en las Reglas 1 y 2 como sigue:

- Regla 1: cada átomo es una fórmula.
- Regla 2: si F_1 y F_2 son fórmulas, entonces también lo son $(F_1 \wedge F_2)$, $(F_1 \vee F_2)$, $\neg(F_1)$ y $\neg(F_2)$. Los valores de verdad de estas fórmulas se derivan de su fórmulas de componentes F_1 y F_2 de la siguiente manera:

¹³ También llamada fórmula bien formada, o WFF, en lógica matemática.

- a. $(F_1 \wedge F_2)$ es VERDADERO si tanto F_1 como F_2 son VERDADEROS ; de lo contrario, es FALSO .
- segundo, $(F_1 \vee F_2)$ es FALSO si tanto F_1 como F_2 son FALSOS ; de lo contrario, es VERDADERO .
- C. $\neg(F_1)$ es TRUE si F_1 es FALSE ; es FALSO si F_1 es VERDADERO .
- re. $\neg(F_2)$ es TRUE si F_2 es FALSE ; es FALSO si F_2 es VERDADERO .

8.6.3 Los cuantificadores existenciales y universales

Además, en las fórmulas pueden aparecer dos símbolos especiales llamados **cuantificadores** ; estas son el **cuantificador universal** (\forall) y el **cuantificador existencial** (\exists). Valores de verdad para las fórmulas con cuantificadores se describen en las Reglas 3 y 4 a continuación; primero, sin embargo, nosotros es necesario definir los conceptos de variables de tupla libres y ligadas en una fórmula. En parámetro, una variable tupla t está limitada si se cuantifica, lo que significa que aparece en un Cláusula ($\exists t$) o ($\forall t$); de lo contrario, es gratis. Formalmente, definimos una variable tupla en una fórmula como **libre** o **ligada** de acuerdo con las siguientes reglas:

- Una ocurrencia de una variable tupla en una fórmula F que es un átomo está libre en F .
- Una ocurrencia de una variable tupla t es libre o está ligada en una fórmula compuesta por conectivos lógicos— $(F_1 \wedge F_2)$, $(F_1 \vee F_2)$, $\neg(F_1)$ y $\neg(F_2)$ — dependiendo de si está libre o ligado en F_1 o F_2 (si ocurre en cualquiera). Observe que en una fórmula de la forma $F = (F_1 \wedge F_2)$ o $F = (F_1 \vee F_2)$, a la variable tupla puede estar libre en F_1 y ligada en F_2 , o viceversa; en este caso, una aparición de la variable tupla está ligada y la otra es libre en F .
- Todas las apariciones libres de una variable tupla t en F están **ligadas** en una fórmula F' de la forma $F' = (\exists t)(F)$ o $F' = (\forall t)(F)$. La variable tupla está vinculada a la cuantificación fier especificado en F' . Por ejemplo, considere las siguientes fórmulas:
 - F_1 : $d.$ Dname = 'Investigación'
 - F_2 : $(\exists t)(d.$ Dnumber = $t.$ Dno)
 - F_3 : $(\forall d)(d.$ Mgr_ssn = '33344555')

La variable tupla d es libre tanto en F_1 como en F_2 , mientras que está ligada a la cuantificación (\forall)

más ardiente en F_3 . La variable t está ligada al cuantificador (\exists) en F_2 .

Ahora podemos dar las Reglas 3 y 4 para la definición de una fórmula que comenzamos antes:

- Regla 3: si F es una fórmula, también lo es $(\exists t)(F)$, donde t es una variable de tupla. la fórmula $(\exists t)(F)$ es VERDADERO si la fórmula F se evalúa como VERDADERO para algunos (al menos uno) tupla asignada a apariciones libres de t en F ; de lo contrario, $(\exists t)(F)$ es FALSO.
- Regla 4: si F es una fórmula, también lo es $(\forall t)(F)$, donde t es una variable tupla. El fórmula $(\forall t)(F)$ es TRUE si la fórmula F se evalúa como TRUE para cada tupla (en el universo) asignado a apariciones libres de t en F ; de lo contrario, $(\forall t)(F)$ es FALSO.

El cuantificador (\exists) se llama cuantificador existencial porque una fórmula $(\exists t)(F)$ es TRUE si existe alguna tupla que haga F TRUE. Para el cuantificador universal, $(\forall t)(F)$ es VERDADERO si todas las tuplas posibles que se pueden asignar a apariciones libres de t en F se sustituye por t , y F es VERDADERO para cada sustitución de este tipo. Se llama el cuantificador universal o para todos porque cada tupla en el universo de tuplas debe haga F TRUE para que la fórmula cuantificada sea TRUE.

8.6.4 Consultas de muestra en cálculo relacional de tuplas

Usaremos algunas de las mismas consultas de la Sección 8.5 para dar una idea de cómo Las mismas consultas se especifican en álgebra relacional y en cálculo relacional. aviso que algunas consultas son más fáciles de especificar en el álgebra relacional que en el relacional cálculo y viceversa.

Consulta 1. Indique el nombre y la dirección de todos los empleados que trabajan para la 'Investigación' Departamento.

P1: $\{t.Fname, t.Lname, t.Address \mid EMPLEADO(t) \text{ Y } (\exists d)(DEPARTAMENTO(d) \text{ AND } d.Dname = 'Investigación' \text{ AND } d.Dnumber = t.Dno)\}$

Las únicas variables de tupla libres en una expresión de cálculo relacional de tuplas deben ser aquellas que aparecen a la izquierda de la barra (\mid). En Q_1 , t es la única variable libre; entonces está atado sucesivamente a cada tupla. Si una tupla satisface las condiciones especificadas después de la barra en Q_1 , los atributos $Fname$, $Lname$ y $Address$ se recuperan para cada una de estas tuplas. los condiciones $EMPLEADO(t)$ y $DEPARTAMENTO(d)$ especifican las relaciones de rango para t y d . La condición $d.Dname = 'Investigación'$ es una **condición de selección** y corresponde a una operación SELECT en el álgebra relacional, mientras que la condición $d.Dnumber = t.Dno$ es una **condición de unión** y tiene un propósito similar al ($INNER$) JOIN funcionamiento (consulte la Sección 8.3).

Consulta 2. Para cada proyecto ubicado en 'Stafford', enumere el número de proyecto, el número del departamento de pesca de curricán y apellido, nacimiento fecha y dirección.

P2: $\{p.Pnumber, p.Dnum, m.Lname, m.Bdate, m.Address \mid PROYECTO(p) \text{ Y } (\exists d)(DEPARTAMENTO(d) \text{ AND } d.Dname = 'Stafford' \text{ AND } d.Dnumber = p.Dnum)\}$

EMPLEADO (m) Y p. Ubicación = 'Stafford' Y $((\exists d) (\text{DEPARTAMENTO} (d)$
 AND p.Dnum = d.Dnumber AND d.Mgr_ssn = m.Ssn))}

En Q2 hay dos variables tuplas libres, py m. La variable de tupla d está vinculada a la cuantificador existencial. La condición de la consulta se evalúa para cada combinación de tuplas asignadas a pym, y de todas las combinaciones posibles de tuplas a las que pym están vinculados, solo se seleccionan las combinaciones que satisfacen la condición.

Varias variables de tupla en una consulta pueden abarcar la misma relación. Por ejemplo, para especificar P8: para cada empleado, recupere el nombre y apellido del empleado y el nombre y apellido de su supervisor inmediato; especificamos dos variables de tupla ables eys que se extienden sobre la relación EMPLEADO :

P8: { e .Fname, e .Lname, s .Fname, s .Lname | EMPLEADO (e) Y EMPLEADO (s)
 Y e .Super_ssn = s .Ssn }

Consulta 3 '. Indique el nombre de cada empleado que trabaja en algún proyecto controlado por departamento número 5. Esta es una variación de Q3 en la que todo se cambia a algunos. En este caso, necesitamos dos condiciones de unión y dos cuantificadores existenciales.

Q0 ' : { e .Lname, e .Fname | EMPLEADO (e) Y $((\exists x) (\exists w) (\text{PROYECTO} (x) \text{ Y}$
 TRABAJA_ON (w) Y x .Dnum = 5 Y w .Essn = e .Ssn Y
 x .Pnumber = w .Pno))}

Consulta 4.Haga una lista de números de proyecto para proyectos que involucran a un empleado cuyo apellido es 'Smith', ya sea como trabajador o como gerente del control departamento para el proyecto.

P4: {p. Pnumber | PROYECTO (p) Y $((\exists e) (\exists w) (\text{EMPLEADO} (e)$
 Y TRABAJA_ON (w) Y w. Pno = p .Pnumber
 Y e .Lname = 'Smith' Y e .Ssn = w .Essn))
 O
 $((\exists m) (\exists d) (\text{EMPLEADO} (m) \text{ Y DEPARTAMENTO} (d)$
 Y p .Dnum = d .Dnumber Y d .Mgr_ssn = m .Ssn
 Y m .Lname = 'Smith'))}

Compare esto con la versión de álgebra relacional de esta consulta en la Sección 8.5. los UNION operación en el álgebra relacional generalmente puede estar sustituido con un O con-nectivo en el cálculo relacional.

8.6.5 Notación para gráficos de consulta

En esta sección, describimos una notación que se ha propuesto para representar relaciones consultas de cálculo que no implican cuantificación compleja en forma gráfica. Estos tipos de consultas se conocen como consultas de **selección-proyecto-unión** porque solo involucran estas tres operaciones de álgebra relacional. La notación se puede ampliar a

consultas más generales, pero no discutimos estas extensiones aquí. Esta gráfica La representación de una consulta se denomina **gráfico de consulta** . La figura 8.13 muestra el gráfico de consulta para Q2 . Las relaciones en la consulta están representadas por **nodos de relación** , que se muestran como círculos simples. Los valores constantes, normalmente de las condiciones de selección de la consulta, son representados por **nodos constantes** , que se muestran como círculos dobles u óvalos. Seleccionando Las condiciones de unión y unión están representadas por los **bordes** del gráfico (las líneas que conectan los nodos), como se muestra en la Figura 8.13. Finalmente, los atributos a recuperar de cada relación se muestran entre corchetes encima de cada relación.

La representación del gráfico de consulta no indica un orden particular para especificar qué operaciones para realizar primero, y por lo tanto es una representación más neutral de una selección consulta de unión de proyecto que la representación del árbol de consultas (consulte la Sección 8.3.5), donde El orden de ejecución se especifica implícitamente. Solo hay un gráfico de consulta respondiendo a cada consulta. Aunque algunas técnicas de optimización de consultas se basaron en los gráficos de consulta, ahora se acepta generalmente que los árboles de consulta son preferibles porque,

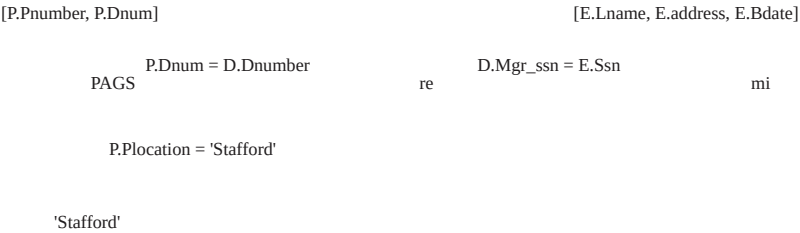


Figura 8.13
Gráfico de consulta para el segundo trimestre.

en la práctica, el optimizador de consultas debe mostrar el orden de las operaciones para la consulta ejecución, que no es posible en los gráficos de consulta.

En la siguiente sección discutimos la relación entre lo universal y lo existencial. cuantificadores y mostrar cómo uno puede transformarse en el otro.

8.6.6 Transformar los cuantificadores universal y existencial

Ahora presentamos algunas transformaciones conocidas de la lógica matemática que relacionar los cuantificadores universal y existencial. Es posible transformar un universal cuantificador en un cuantificador existencial, y viceversa, para obtener una expresión equivalente. Una transformación general puede describirse informalmente como sigue: formar un tipo de cuantificador en otro con negación (precedido de NOT); Y y O reemplazarse entre sí; una fórmula negada se vuelve innecesaria; y un la fórmula negada se vuelve negada. Algunos casos especiales de esta transformación pueden ser se indica de la siguiente manera, donde el símbolo \equiv significa **equivalente a** :

$$\begin{aligned}
 (\forall x) (P(x)) &\equiv \text{NO} (\exists x) (\text{NO} (P(x))) \\
 (\forall x) (P(x) \vee Q(x)) &\equiv \text{NO} (\exists x) (\text{NO} (P(x)) \wedge \text{NO} (Q(x))) \\
 (\forall x) (P(x) \wedge Q(x)) &\equiv \text{NO} (\exists x) (\text{NO} (P(x)) \vee \text{NO} (Q(x))) \\
 (\exists x) (P(x) \wedge Q(x)) &\equiv \text{NO} (\forall x) (\text{NO} (P(x)) \vee \text{NO} (Q(x))) \\
 (\exists x) (P(x) \vee Q(x)) &\equiv \text{NO} (\forall x) (\text{NO} (P(x)) \wedge \text{NO} (Q(x)))
 \end{aligned}$$

Observe también que el siguiente es VERDADERO , donde el símbolo \Rightarrow significa **implica** :

$$\begin{aligned}
 (\forall x) (P(x)) &\Rightarrow (\exists x) (P(x)) \\
 \text{NO} (\exists x) (P(x)) &\Rightarrow \text{NO} (\forall x) (P(x))
 \end{aligned}$$

8.6.7 Uso del cuantificador universal en consultas

Siempre que utilizamos un cuantificador universal, es bastante sensato seguir algunas reglas para asegurarse de que nuestra expresión tenga sentido. Discutimos estas reglas con respecto a la consulta Q3 .

Consulta 3. Enumere los nombres de los empleados que trabajan en todos los proyectos controlados por número de departamento 5. Una forma de especificar esta consulta es utilizar la cuantificador como se muestra:

P3: {e .Lname, e .Fname | EMPLEADO (e) Y (($\forall x$) (NO (PROYECTO (x)) O NO (x .Dnum = 5) O (($\exists w$) (WORKS_ON (w) AND w .Essn = e .Ssn AND x .Pnumber = w .Pno))))}

Podemos dividir Q3 en sus componentes básicos de la siguiente manera:

P3: {e .Lname, e .Fname | EMPLEADO (e) Y F ' }
 $F' = ((\forall x) (\text{NO} (\text{PROYECTO} (x)) \vee F_1))$
 $F_1 = \text{NO} (x .Dnum = 5) \vee F_2$
 $F_2 = ((\exists w) (\text{TRABAJO_ON} (w) \vee w .Essn = e .Ssn \vee x .Pnumber = w .Pno))$

Queremos asegurarnos de que un empleado seleccionado trabaje en todos los proyectos controlado por el departamento 5, pero la definición de cuantificador universal dice que para hacer que la fórmula cuantificada sea VERDADERA , la fórmula interna debe ser VERDADERA para todas las tuplas En el universo. El truco consiste en excluir de la cuantificación universal todas las tuplas. que no nos interesa al hacer que la condición sea VERDADERA para todas esas tuplas. Esto es necesario porque una variable de tupla cuantificada universalmente, como x en Q3 , debe evaluar a VERDADERO para cada posible tupla asignada para hacer el cuantificado fórmula VERDADERO .

Las primeras tuplas para excluir (al hacer que se evalúen automáticamente como VERDADERO) son los que no están en la relación R de interés. En el tercer trimestre , usando la expresión $\text{NO} (\text{PROYECTO} (x))$ dentro de la fórmula cuantificada universalmente se evalúa como VERDADERO todos tuplas x que no están en la relación PROYECTO . Entonces excluimos las tuplas que no somos

interesado en la propia R. En Q3, el uso de la expresión NOT (x. Dnum = 5) se evalúa como VERDADERO todas las tuplas x que están en la relación PROYECTO pero que no están controladas por departamentos. ment 5. Finalmente, especificamos una condición F₂ que debe mantenerse en todas las tuplas restantes en R. Por lo tanto, podemos explicar Q3 de la siguiente manera:

1. Para que la fórmula $F' = (\forall x) (F)$ sea VERDADERA, debemos tener la fórmula F VERDADERO para todas las tuplas del universo que se pueden asignar a x. Sin embargo, en el tercer trimestre solo nos interesa que F sea VERDADERO para todas las tuplas de la relación PROYECTO que están controlados por el departamento 5. Por lo tanto, la fórmula F tiene la forma (NO (PROYECTO (x)) O F₁). La condición ' NO (PROYECTO (x)) O ...' es VERDADERO para todas las tuplas que no están en la relación PROYECTO y tiene el efecto de eliminar derivando estas tuplas de la consideración en el valor de verdad de F₁. Para cada tupla en la relación PROYECTO, F₁ debe ser VERDADERO si F' debe ser VERDADERO.
2. Usando la misma línea de razonamiento, no queremos considerar tuplas en el Relación PROYECTO que no están controlados por el departamento número 5, ya que solo están interesados en tuplas PROYECTO cuyo Dnum = 5. Por tanto, podemos escribir:

$$SI (x.Dnum = 5) ENTONCES F_2$$
 que es equivalente a

$$(NO (x.Dnum = 5) O F_2)$$
3. La fórmula F₁, por tanto, tiene la forma NOT (x. Dnum = 5) OR F₂. En el contexto de Q3, esto significa que, para una tupla x en la relación PROYECTO, su Dnum ≠ 5 o debe satisfacer F₂.
4. Finalmente, F₂ da la condición que queremos mantener para un EMPLEADO seleccionado tupla: que el empleado trabaja en cada tupla de PROYECTO que no ha sido excluido todavía. La consulta selecciona dichas tuplas de empleados.

En inglés, Q3 da la siguiente condición para seleccionar una tupla e EMPLEADO :
 Para cada tupla x en la relación PROYECTO con x. Dnum = 5, debe existir una tupla w en WORKS_ON tal que w. Essn = e. Ssn y w. Pno = x. Pnumber. Esto es equivalente a decir que EMPLEADO e trabaja en cada PROYECTO x en el DEPARTAMENTO número 5. (¡Uf!)

Usando la transformación general de cuantificadores universales a existenciales dada en la sección 8.6.6, podemos reformular la consulta en Q3 como se muestra en Q3A, que usa un cuantificador existencial negado en lugar del cuantificador universal:

$$Q3A: \{ e.Lname, e.Fname \mid EMPLEADO(e) \wedge (\text{NO} (\exists x) (PROYECTO(x) \wedge (x.Dnum = 5)) \wedge (\text{NO} (\exists w) (WORKS_ON(w) \wedge w.Essn = e.Ssn \wedge x.Pnumber = w.Pno)))) \}$$

A continuación, ofrecemos algunos ejemplos adicionales de consultas que utilizan cuantificadores.

Consulta 6. Enumere los nombres de los empleados que no tienen dependientes.

P6: $\{e.Fname, e.Lname \mid EMPLEADO(e) \wedge (\neg \exists d) (DEPENDIENTE(d) \wedge e.Ssn = d.Essn)\}$

Usando la regla de transformación general, podemos reformular Q6 de la siguiente manera:

P6A: $\{e.Fname, e.Lname \mid EMPLEADO(e) \wedge ((\forall d) (\neg (DEPENDIENTE(d) \vee \neg (e.Ssn = d.Essn))))\}$

Consulta 7. Enumere los nombres de los gerentes que tienen al menos un dependiente.

P7: $\{e.Fname, e.Lname \mid EMPLEADO(e) \wedge ((\exists d) (\exists p) (DEPARTAMENTO(d) \wedge DEPENDIENTE(p) \wedge e.Ssn = d.Mgr_ssn \wedge p.Essn = e.Ssn))\}$

Esta consulta se maneja interpretando a los gerentes que tienen al menos un dependiente como gerentes para quienes existe algún dependiente.

8.6.8 Expresiones seguras

Siempre que usamos cuantificadores universales, cuantificadores existenciales o negación de predicates en una expresión de cálculo, debemos asegurarnos de que la expresión resultante tiene sentido. Una **expresión segura** en el cálculo relacional es aquella que garantiza producir un número finito de tuplas como resultado; de lo contrario, la expresión se denomina **insegura**. Por ejemplo, la expresión

$\{t \mid \neg (EMPLEADO(t))\}$

no es segura porque produce todas las tuplas del universo que no son tuplas EMPLEADOS, que son infinitamente numerosos. Si seguimos las reglas para el tercer trimestre discutidas anteriormente, obtendrá una expresión segura cuando utilice cuantificadores universales. Podemos definir seguro Expresiones con mayor precisión al introducir el concepto de dominio de una tupla. expresión de cálculo relacional: este es el conjunto de todos los valores que aparecen como valores constantes en la expresión o existen en cualquier tupla en las relaciones referenciadas en la expresión. Por ejemplo, el dominio de $\{t \mid \neg (EMPLOYEE(t))\}$ es el conjunto de todos los valores de atributo que aparecen en alguna tupla de la relación EMPLEADO (para cualquier atributo). El dominio de la expresión Q3A incluiría todos los valores que aparecen en EMPLOYEE, PROJECT y WORKS_ON (unido con el valor 5 que aparece en la consulta en sí).

Se dice que una expresión es **segura** si todos los valores en su resultado son del dominio de la expresión. Observe que el resultado de $\{t \mid \neg (EMPLOYEE(t))\}$ no es seguro, ya que lo hará,

8.7 El dominio del cálculo relacional

Existe otro tipo de cálculo relacional llamado cálculo relacional de dominio, o simplemente **cálculo de dominio**. Históricamente, mientras que SQL (véanse los capítulos 6 y 7), basado en el cálculo relacional de tuplas, estaba siendo desarrollado por IBM Research en San José, California, otro idioma llamado QBE (Query-By-Example), que es relacionado con el cálculo de dominios, se estaba desarrollando casi al mismo tiempo en IBM Centro de Investigación TJ Watson en Yorktown Heights, Nueva York. La especificación formal del cálculo de dominio se propuso después del desarrollo del lenguaje QBE calibre y sistema.

El cálculo de dominios difiere del cálculo de tuplas en el tipo de variables utilizadas en la fórmula. En lugar de tener un rango de variables en tuplas, las variables varían en un solo valores de dominios de atributos. Para formar una relación de grado n para el resultado de una consulta, debemos tener n de estas **variables de dominio**, una para cada atributo. Una expresión del cálculo de dominio es de la forma

$$\{x_1, x_2, \dots, x_n \mid \text{COND}(x_1, x_2, \dots, x_n, x_{n+1}, x_{n+2}, \dots, x_{n+m})\}$$

donde $x_1, x_2, \dots, x_n, x_{n+1}, x_{n+2}, \dots, x_{n+m}$ son variables de dominio que van más allá de dominios (de atributos), y COND es una **condición** o **fórmula** del dominio cálculo relacional.

Una fórmula está formada por **átomos**. Los átomos de una fórmula son ligeramente diferentes de los del cálculo de tuplas y puede ser uno de los siguientes:

1. Un átomo de la forma $R(x_1, x_2, \dots, x_j)$, donde R es el nombre de una relación de grado j y cada x_i , $1 \leq i \leq j$, es una variable de dominio. Este átomo afirma que una lista de valores de $\langle x_1, x_2, \dots, x_j \rangle$ debe ser una tupla en la relación cuyo nombre es R , donde x_i es el valor del i -ésimo valor de atributo de la tupla. Hacer un dominio expresión de cálculo más concisa, podemos quitar las comas en una lista de valores; así, podemos escribir:

$$\{x_1, x_2, \dots, x_n \mid R(x_1 x_2 x_3) \vee \dots\}$$

en lugar de:

$$\{x_1, x_2, \dots, x_n \mid R(x_1, x_2, x_3) \vee \dots\}$$

2. Un átomo de la forma $x_i \text{ op } x_j$, donde **op** es uno de los operadores de comparación en el conjunto $\{=, <, \leq, >, \geq, \neq\}$, y x_i y x_j son variables de dominio.
3. Un átomo de la forma $x_i \text{ op } c \text{ o } c \text{ op } x_j$, donde **op** es uno de los operadores en el conjunto $\{=, <, \leq, >, \geq, \neq\}$, x_i y x_j son variables de dominio, y c es un valor constante.

Como en el cálculo de tuplas, los átomos se evalúan como VERDADERO o FALSO para un conjunto específico de valores, llamados **valores de verdad** de los átomos. En el caso 1, si las variables de dominio son

asignados valores correspondientes a una tupla de la relación especificada R, entonces el átomo es VERDADERO . En los casos 2 y 3, si a las variables de dominio se les asignan valores que satisfacen condición, entonces el átomo es VERDADERO .

De manera similar al cálculo relacional de tuplas, las fórmulas se componen de átomos, variables y cuantificadores, por lo que no repetiremos las especificaciones para fórmulas aquí. A continuación se muestran algunos ejemplos de consultas especificadas en el cálculo del dominio. Usaremos bajo ercasas letras l, m, n, ..., x, y, z para variables de dominio.

Consulta 0. Indique la fecha de nacimiento y la dirección del empleado cuyo nombre es 'John B. Smith ».

$$Q0: \{u, v \mid (\exists q) (\exists r) (\exists s) (\exists t) (\exists w) (\exists x) (\exists y) (\exists z) \\ (\text{EMPLEADO} (qrstuvwxyz) \text{ AND } q = \text{'John'} \text{ AND } r = \text{'B'} \text{ AND } s = \text{'Smith'})\}$$

Necesitamos diez variables para la relación EMPLEADO , una para abarcar cada una de las dominios de atributos del EMPLEADO en orden. De las diez variables q, r, s, ..., z, sólo u y v son libres, porque aparecen a la izquierda de la barra y por lo tanto deberían no estar ligado a un cuantificador. Primero especificamos los atributos solicitados, Bdate y Dirección , por las variables de dominio libres u para BDATE y v para DIRECCIÓN . Entonces nosotros especificar la condición para seleccionar una tupla después de la barra (|), es decir, que el secuencia de valores asignados a las variables qrstuvwxyz ser una tupla del EMPLEADO relación y que los valores de q (Fnombre), r (Minit) y s (Lnombre) sean iguales a 'John', 'B' y 'Smith', respectivamente. Por conveniencia, cuantificaremos solo aquellos variables que realmente aparecen en una condición (estas serían q, r y s en Q0) en el resto de nuestros ejemplos. ¹⁴

Una notación abreviada alternativa, utilizada en QBE, para escribir esta consulta es asignar las constantes 'John', 'B' y 'Smith' directamente como se muestra en Q0A . Aquí, todas las variables que no aparecen a la izquierda de la barra están implícitamente cuantificados existencialmente: ¹⁵

$$Q0A: \{u, v \mid \text{EMPLEADO} (\text{'John'}, \text{'B'}, \text{'Smith'}, t, u, v, w, x, y, z)\}$$

Consulta 1. Obtenga el nombre y la dirección de todos los empleados que trabajan para 'Departamento de Investigación.

$$P1: \{q, s, v \mid (\exists z) (\exists l) (\exists m) (\text{EMPLEADO} (qrstuvwxyz) \text{ Y} \\ \text{DEPARTAMENTO} (lmno) \text{ AND } l = \text{'Investigación'} \text{ AND } m = z)\}$$

Una condición que relaciona dos variables de dominio que abarcan atributos de dos relaciones, como m = z en Q1 , es una **condición de unión** , mientras que una condición que relaciona una variable de dominio a una constante, como l = 'Investigación', es una **condición de selección** .

Consulta 2. Para cada proyecto ubicado en 'Stafford', enumere el número de proyecto, el número del departamento de pesca de curricán y apellido, nacimiento fecha y dirección.

¹⁴ Cuantificar solo las variables de dominio realmente utilizadas en las condiciones y especificar un predicado como EMPLOYEE (qrstuvwxyz) sin separar las variables de dominio con comas es una notación abreviada utilizado por conveniencia; no es la notación formal correcta.

¹⁵ Nuevamente, esta no es una notación formalmente precisa.

P2: $\{i, k, s, u, v \mid (\exists j) (\exists m) (\exists n) (\exists t) (\text{PROYECTO} (hijk) \text{ Y } \text{EMPLEADOS} (QRSTUVWXYZ) \text{ DEPARTAMENTO DE } (LMNO) \text{ Y } k = m \text{ Y } n = t \text{ Y } j = \text{'Stafford'}) \}$

Consulta 6. Enumere los nombres de los empleados que no tienen dependientes.

P6: $\{q, s \mid (\exists t) (\text{EMPLEADO} (qrstuvwxyz) \text{ Y } (\text{NO } (\exists l) (\text{DEPENDIENTE} (lmnop) \text{ Y } t = l))) \}$

Q6 se puede reformular utilizando cuantificadores universales en lugar de cuantificadores existenciales, como se muestra en Q6A :

P6A: $\{q, s \mid (\exists t) (\text{EMPLEADO} (qrstuvwxyz) \text{ Y } ((\forall l) (\text{NO } (\text{DEPENDIENTE} (lmnop)) \text{ O NO } (t = l)))) \}$

Consulta 7. Enumere los nombres de los gerentes que tienen al menos un dependiente.

P7: $\{s, q \mid (\exists t) (\exists j) (\exists l) (\text{EMPLEADO} (qrstuvwxyz) \text{ Y DEPARTAMENTO } (hijk) \text{ Y DEPENDIENTE } (lmnop) \text{ Y } t = j \text{ Y } l = t) \}$

Como mencionamos anteriormente, se puede demostrar que cualquier consulta que se pueda expresar en el álgebra relacional básica también se puede expresar en el dominio o tupla relacional cálculo. Además, cualquier expresión segura en el dominio o tupla de cálculo relacional puede ser expresado en el álgebra relacional básica.

El lenguaje QBE se basó en el cálculo relacional de dominio, aunque esto fue se dio cuenta más tarde, después de que se formalizara el cálculo de dominios. QBE fue uno de los primeros Lenguajes de consulta gráfica con sintaxis mínima desarrollados para sistemas de bases de datos. Eso fue desarrollado en IBM Research y está disponible como un producto comercial de IBM como parte de la opción de interfaz Query Management Facility (QMF) para DB2. Lo básico Las ideas utilizadas en QBE se han aplicado en varios otros productos comerciales. Porque de su importante lugar en la historia de los lenguajes relacionales, hemos incluido un descripción general de QBE en el Apéndice C.

8.8 Resumen

En este capítulo presentamos dos lenguajes formales para el modelo relacional de datos. Se utilizan para manipular relaciones y producir nuevas relaciones como respuestas a preguntas. Ries. Discutimos el álgebra relacional y sus operaciones, que se utilizan para especificar ify una secuencia de operaciones para especificar una consulta. Luego presentamos dos tipos de cálculos relacionales llamados cálculo de tuplas y cálculo de dominios.

En las secciones 8.1 a 8.3, presentamos las operaciones básicas de álgebra relacional e ilustró los tipos de consultas para las que se utiliza cada una. Primero, discutimos el los operadores relacionales unarios SELECT y PROJECT , así como la operación RENAME . Luego, discutimos las operaciones teóricas de conjuntos binarios que requieren que las relaciones en que se apliquen sean compatibles con la unión (o tipo); estos incluyen UNION , INTERSECTION y SET DIFFERENCE . La operación PRODUCTO CARTESIANO es una operación de conjunto que se puede usar para combinar tuplas de dos relaciones, produciendo todas las combinaciones posibles. Rara vez se usa en la práctica; sin embargo, mostramos cómo

PRODUCTO CARTESIANO seguido de SELECCIONAR se puede utilizar para definir tuplas coincidentes de dos relaciones y conduce a la operación JOIN . Diferentes operaciones JOIN llamadas

Se introdujeron THETA JOIN , EQUIJOIN y NATURAL JOIN . Se introdujeron árboles de consulta generados como una representación gráfica de consultas de álgebra relacional, que también se puede utilizar como base para las estructuras de datos internas que el DBMS puede utilizar para representar una consulta.

Discutimos algunos tipos importantes de consultas que no se pueden establecer con las operaciones de álgebra relacional pero son importantes para situaciones prácticas. Introducimos PROYECCIÓN GENERALIZADA para utilizar funciones de atributos en la proyección lista y la operación FUNCIÓN AGREGADA para tratar con tipos agregados de estadísticas solicitadas que resumen la información en las tablas. Discutimos consultas recursivas, para las cuales no hay apoyo directo en el álgebra pero que pueden ser manejadas de manera recursiva en un enfoque paso a paso, como demostramos. Luego presentamos el EXTERIOR

Operaciones JOIN y OUTER UNION , que amplían JOIN y UNION y permiten información en las relaciones de fuente que se conservará en el resultado.

Las dos últimas secciones describieron los conceptos básicos detrás del cálculo relacional, que se basa en la rama de la lógica matemática denominada cálculo de predicados. Existen dos tipos de cálculos relacionales: (1) el cálculo relacional de tuplas, que usa tuplas variables que se extienden sobre tuplas (filas) de relaciones, y (2) el dominio relacional cálculo, que utiliza variables de dominio que se extienden sobre dominios (columnas de relaciones). En el cálculo relacional, una consulta se especifica en una sola declaración declarativa, sin especificar ningún orden o método para recuperar el resultado de la consulta. Por tanto, el cálculo relacional se considera a menudo como un lenguaje declarativo de nivel superior al álgebra relacional, porque una expresión de cálculo relacional establece lo que queremos recuperar independientemente de cómo se pueda ejecutar la consulta.

Introducimos gráficos de consulta como una representación interna para consultas en relaciones de cálculo. También discutimos el cuantificador existencial (\exists) y el cuantificador universal (\forall). Discutimos el problema de especificar consultas seguras cuyos resultados son finitos. También discutimos las reglas para transformar lo universal en cuantificación existencial, y viceversa. Son los cuantificadores los que dan poder expresivo a la relación de cálculo, haciéndolo equivalente al álgebra relacional básica. No hay análogo a funciones de agrupamiento y agregación en el cálculo relacional básico, aunque algunas se han sugerido extensiones.

Preguntas de revisión

- 8.1. Enumere las operaciones del álgebra relacional y el propósito de cada una.
- 8.2. ¿Qué es la compatibilidad sindical? ¿Por qué la UNIÓN , INTERSECCIÓN y las operaciones de DIFERENCIA requieren que las relaciones sobre las que se aplicado ser compatible con la unión?
- 8.3. Analizar algunos tipos de consultas para las que es necesario cambiar el nombre de los atributos. para especificar la consulta de forma inequívoca.
- 8.4. Analice los distintos tipos de operaciones de combinación interna. ¿Por qué se requiere la unión theta?

- 8.5. ¿Qué papel juega el concepto de clave externa al especificar la mayoría tipos comunes de operaciones de combinación significativas?
- 8.6. ¿Qué es la operación FUNCIÓN ? ¿Para que se usa?
- 8.7. ¿Cómo son los OUTER JOIN operaciones diferentes de la combinación interna operaciones? ¿En qué se diferencia la operación OUTER UNION de UNION ?
- 8.8. ¿En qué sentido el cálculo relacional difiere del álgebra relacional, y en que sentido son similares?
- 8.9. ¿En qué se diferencia el cálculo relacional de tuplas del cálculo relacional de dominios?
- 8.10. Discuta los significados del cuantificador existencial (\exists) y el universal cuantificador (\forall).
- 8.11. Defina los siguientes términos con respecto al cálculo de tuplas: variable de tuplas, relación de rango, átomo, fórmula y expresión.
- 8.12. Defina los siguientes términos con respecto al cálculo de dominio: dominio variable, relación de rango, átomo, fórmula y expresión.
- 8.13. ¿Qué se entiende por expresión segura en cálculo relacional?
- 8.14. ¿Cuándo se llama relacionalmente completo un lenguaje de consulta?

Ejercicios

- 8.15. Muestre el resultado de cada una de las consultas de muestra en la Sección 8.5 como se aplicaría al estado de la base de datos en la Figura 5.6.
- 8.16. Especifique las siguientes consultas en el esquema de la base de datos relacional de la EMPRESA que se muestra en la Figura 5.5 utilizando los operadores relacionales discutidos en este capítulo. También muestre el resultado de cada consulta, ya que se aplicaría al estado de la base de datos en Figura 5.6.
 - a. Recuperar los nombres de todos los empleados del departamento 5 que trabajan más de 10 horas por semana en el proyecto ProductX.
 - segundo. Enumere los nombres de todos los empleados que tienen un dependiente con el mismo primero nombre como ellos mismos.
 - C. Busque los nombres de todos los empleados supervisados directamente por 'Franklin Wong'.
 - re. Para cada proyecto, enumere el nombre del proyecto y el total de horas por semana (por empleados) gastados en ese proyecto.
 - mi. Recupera los nombres de todos los empleados que trabajan en cada proyecto.
 - F. Recupera los nombres de todos los empleados que no trabajan en ningún proyecto.
 - gramo. Para cada departamento, recupere el nombre del departamento y el saldo promedio ario de todos los empleados que trabajan en ese departamento.
 - h. Recupere el salario medio de todas las empleadas.

- yo. Busque los nombres y direcciones de todos los empleados que trabajan en al menos una proyecto ubicado en Houston pero cuyo departamento no tiene ubicación en Houston.
- j. Enumere los apellidos de todos los directores de departamento que no tengan dependientes.
- 8.17. Considere el esquema de la base de datos relacional AIRLINE que se muestra en la Figura 5.8, que se describió en el ejercicio 5.12. Especifique las siguientes consultas en relacional álgebra:
- a. Para cada vuelo, indique el número de vuelo, el aeropuerto de salida para el primer tramo del vuelo y el aeropuerto de llegada para el último tramo del vuelo.
- segundo. Indique los números de vuelo y los días de la semana de todos los vuelos o tramos de vuelo que salen desde el Aeropuerto Intercontinental de Houston (código de aeropuerto 'iah') y llegar a Aeropuerto Internacional de Los Ángeles (código de aeropuerto 'lax').
- C. Indique el número de vuelo, el código del aeropuerto de salida, la hora de salida programada, código de aeropuerto de llegada, hora de llegada programada y días de semana de todos los vuelos o tramos de vuelo que parten de algún aeropuerto de la ciudad de Houston y llegan en algún aeropuerto de la ciudad de Los Ángeles.
- re. Enumere toda la información de tarifas para el número de vuelo 'co197'.
- mi. Recupere el número de asientos disponibles para el número de vuelo 'co197' en '2009-10-09'.
- 8.18. Considere el esquema de la base de datos relacional LIBRARY que se muestra en la Figura 8.14, que se utiliza para realizar un seguimiento de los libros, los prestatarios y los préstamos de libros. Integridad referencial Las restricciones se muestran como arcos dirigidos en la figura 8.14, como en la notación de la figura. ure 5.7. Escriba expresiones relacionales para las siguientes consultas:
- a. ¿Cuántas copias del libro titulado The Lost Tribe son propiedad de la sucursal de la biblioteca cuyo nombre es 'Sharpstown'?
- segundo. ¿Cuántas copias del libro titulado The Lost Tribe tiene cada uno? sucursal de la biblioteca?
- C. Recuperar los nombres de todos los prestatarios que no tienen libros. controlado.
- re. Por cada libro prestado desde la sucursal de Sharpstown y cuyo Due_date es hoy, recupere el título del libro, el nombre del prestatario y el dirección del prestatario.
- mi. Para cada sucursal de la biblioteca, recupere el nombre de la sucursal y el número total de libros prestados desde esa sucursal.
- F. Recuperar los nombres, direcciones y la cantidad de libros prestados para todos prestatarios que tienen más de cinco libros prestados.
- gramo. Para cada libro escrito (o coautor) por Stephen King, recupere el título y el número de copias propiedad de la sucursal de la biblioteca cuyo nombre es Central.
- 8.19. Especifique las siguientes consultas en álgebra relacional en el esquema de la base de datos dado en el ejercicio 5.14:

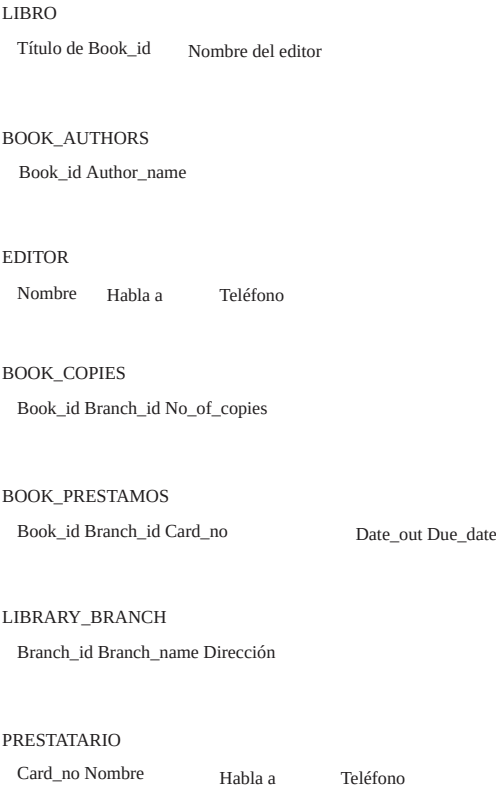


Figura 8.14
Una base de datos relacional
esquema para una BIBLIOTECA
base de datos.

- a. Indique el número de pedido y la fecha de envío de todos los pedidos enviados desde el almacén número W2.
- segundo. Enumere la información de ALMACÉN de la que el CLIENTE
llamado José López recibió sus órdenes. Producir un listado: Orden # ,
Almacén # .
- C. Producir un listado Cname , No_of_orders , Avg_order_amt , donde el medio
la columna es el número total de pedidos del cliente y la última columna
es el importe medio del pedido de ese cliente.
- re. Enumere los pedidos que no se enviaron dentro de los 30 días posteriores al pedido.
- mi. Indique el número de pedido para los pedidos que se enviaron desde todos los almacenes que
empresa tiene en Nueva York.

8.20. Especifique las siguientes consultas en álgebra relacional en el esquema de la base de datos dado en el ejercicio 5.15:

- a. Proporcione los detalles (todos los atributos de la relación de viaje) para viajes que excedieron \$ 2,000 en gastos.

segundo. Imprima las Ssn de los vendedores que viajaron a Honolulu.

C. Imprima los gastos de viaje totales incurridos por el vendedor con SSN = «234-56-7890».

8.21. Especifique las siguientes consultas en álgebra relacional en el esquema de la base de datos dado en el ejercicio 5.16:

- a. Enumere el número de cursos tomados por todos los estudiantes llamados John Smith en Invierno de 2009 (es decir, trimestre = S09).

segundo. Produzca una lista de libros de texto (incluya Course # , Book_isbn , Book_title) para cursos ofrecidos por el departamento de 'CS' que han utilizado más de dos libros.

C. Enumere cualquier departamento que tenga todos sus libros adoptados publicados por 'Pearson Publicación'.

8.22. Considere las dos tablas T1 y T2 que se muestran en la figura 8.15. Mostrar los resultados de las siguientes operaciones:

a. $T1 \bowtie_{T1.P = T2.A} T2$

segundo. $T1 \bowtie_{T1.Q = T2.B} T2$

C. $T1 \bowtie_{T1.P = T2.A} T2$

re. $T1 \bowtie_{T1.Q = T2.B} T2$

mi. $T1 \cup T2$

F. $T1 \bowtie_{(T1.P = T2.A \vee T1.R = T2.C)} T2$

8.23. Especifique las siguientes consultas en álgebra relacional en el esquema de la base de datos en Ejercicio 5.17:

- a. Para el vendedor llamado 'Jane Doe', enumere la siguiente información para todos los coches que vendió: número de serie , fabricante , precio de venta .

segundo. Enumere el número de serie y el modelo de los automóviles que no tienen opciones.

C. Considere la operación NATURAL JOIN entre VENDEDOR y VENTA . ¿Cuál es el significado de una combinación externa izquierda para estas tablas (no cambiar el orden de las relaciones)? Explique con un ejemplo.

re. Escriba una consulta en álgebra relacional que involucre selección y un conjunto de operaciones y decir con palabras lo que hace la consulta.

8.24. Especifique las consultas a, b, c, e, f, i y j del ejercicio 8.16 tanto en la tupla como en el dominio cálculo relacional.

8.25. Especifique las consultas a, b, c y d del ejercicio 8.17 tanto en la relación de tuplas como de dominios.

Figura 8.15
Un estado de base de datos para el
relaciones T1 y T2.

TABLA T1			TABLA T2		
PAGS	Q	R	UN	segundo	C
10	un	5	10	segundo	6
15	segundo	8	25	C	3
25	un	6	10	segundo	5

- 8.26. Especifique las consultas c, d y f del ejercicio 8.18 tanto en relación de tuplas como de dominios.
- 8.27. En una consulta de cálculo relacional de tuplas con n variables de tuplas, ¿cuál sería la número mínimo típico de condiciones de unión? ¿Por qué? Cual es el efecto de ¿Tiene un número menor de condiciones de unión?
- 8.28. Reescriba las consultas de cálculo relacional de dominio que siguieron a Q0 en la sección ción 8.7 en el estilo de la notación abreviada de Q0A , donde el objetivo es minimizar el número de variables de dominio escribiendo constantes en su lugar de variables siempre que sea posible.
- 8.29. Considere esta consulta: recupere los SSN de los empleados que trabajan en al menos aquellos proyectos en los que trabaja el empleado con Ssn = 123456789. Esto puede expresarse como (FORALL x) (IF P THEN Q), donde
- x es una variable de tupla que se extiende sobre la relación PROYECTO .
 - P ≡ empleado con Ssn = 123456789 trabaja en el proyecto x.
 - Q ≡ empleado e trabaja en el proyecto x.
- Expresa la consulta en cálculo relacional de tuplas, usando las reglas
- $(\forall x) (P(x)) \equiv \text{NO} (\exists x) (\text{NO} (P(x)))$.
 - $(\text{SI } P \text{ ENTONCES } Q) \equiv (\text{NO} (P) \text{ O } Q)$.
- 8.30. Muestre cómo puede especificar las siguientes operaciones de álgebra relacional en cálculo relacional de tuplas y dominios.
- a. $\sigma_A = C (R (A, B, C))$
 - segundo. $\pi_{\langle A, B \rangle} (R (A, B, C))$
 - C. $R (A, B, C) * S (C, D, E)$
 - re. $R (A, B, C) \cup S (A, B, C)$
 - mi. $R (A, B, C) \cap S (A, B, C)$
 - F. $R (A, B, C) = S (A, B, C)$
 - gramo. $R (A, B, C) \times S (D, E, F)$

h. $R(A, B) \div S(A)$

- 8.31. Sugerir extensiones al cálculo relacional para que pueda expresar lo siguiente tipos de operaciones que se discutieron en la Sección 8.4: (a) funciones de puerta y agrupación; (b) operaciones OUTER JOIN ; (c) recursivo consultas de cierre.
- 8.32. Una consulta anidada es una consulta dentro de una consulta. Más específicamente, una consulta anidada es una consulta entre paréntesis cuyo resultado se puede utilizar como un valor en una serie de lugares, como en lugar de una relación. Especifique las siguientes consultas en el base de datos especificada en la Figura 5.5 utilizando el concepto de consultas anidadas y la operadores relacionales discutidos en este capítulo. También muestre el resultado de cada consulta como se aplicaría al estado de la base de datos en la Figura 5.6.
- a. Enumere los nombres de todos los empleados que trabajan en el departamento que tiene empleado con el salario más alto entre todos los empleados.

segundo. Enumere los nombres de todos los empleados cuyo supervisor haya '888665555' para S_{sn} .

C. Enumere los nombres de los empleados que ganan al menos \$ 10,000 más que el empleado al que se le paga menos en la empresa.

8.33. Indique si las siguientes conclusiones son verdaderas o falsas:

a. $\text{NO} (P(x) \text{ O } Q(x)) \rightarrow (\text{NO} (P(x)) \text{ Y } (\text{NO} (Q(x))))$

segundo. $\text{NO} (\exists x) (P(x)) \rightarrow \forall x (\text{NO} (P(x)))$

C. $(\exists x) (P(x)) \rightarrow \forall x ((P(x))$

Ejercicios de laboratorio

- 8.34. Especifique y ejecute las siguientes consultas en álgebra relacional (RA) usando el intérprete RA en el esquema de la base de datos EMPRESA en la Figura 5.5.
- a. Enumere los nombres de todos los empleados del departamento 5 que trabajan más de 10 horas por semana en el proyecto ProductX.
- segundo. Enumere los nombres de todos los empleados que tienen un dependiente con el mismo primero nombre como ellos mismos.
- C. Enumere los nombres de los empleados supervisados directamente por Franklin Wong.
- re. Enumere los nombres de los empleados que trabajan en cada proyecto.
- mi. Enumere los nombres de los empleados que no trabajan en ningún proyecto.
- F. Enumere los nombres y direcciones de los empleados que trabajan en al menos una proyecto ubicado en Houston pero cuyo departamento no tiene ubicación en Houston.
- gramo. Enumere los nombres de los gerentes de departamento que no tienen dependientes.

8.35. Considere el siguiente esquema relacional MAILORDER que describe los datos para una empresa de venta por correo.

PIEZAS (Pno, Pname, Qoh, Price, Olevel)
 CLIENTES (Cno, Cname, Street, Zip, Phone)
 EMPLEADOS (Eno, Ename, Zip, Hdate)
 ZIP_CODES (código postal, ciudad)
 PEDIDOS (Ono, Cno, Eno, Recibidos, Enviados)
 DETALLES (Ono, Pno, Cant.)

Qoh significa cantidad disponible; los otros nombres de atributos son explicativo. Especifique y ejecute las siguientes consultas utilizando el RA intérprete en el esquema de la base de datos MAILORDER .

- a. Recupere los nombres de las piezas que cuestan menos de \$ 20,00.
- segundo. Recupere los nombres y las ciudades de los empleados que han recibido pedidos piezas que cuestan más de \$ 50,00.
- C. Recupere los pares de valores de número de cliente de los clientes que viven en el mismo código postal.

re. Recuperar los nombres de los clientes que han pedido piezas a los empleados. viviendo en Wichita.

mi. Recupere los nombres de los clientes que han pedido piezas que cuestan menos de \$ 20,00.

F. Recupere los nombres de los clientes que no han realizado un pedido.

gramo. Recupere los nombres de los clientes que han realizado exactamente dos pedidos.

8.36. Considere el siguiente esquema relacional GRADEBOOK que describe los datos para un libro de calificaciones de un instructor en particular. (Nota: Los atributos A , B , C , y D de los límites de grado de la tienda de CURSOS).

CATÁLOGO (Cno, Ctitle)
 ESTUDIANTES (Sid, Fname, Lname, Minit)
 CURSOS (Término, Sec_no, Cno, A, B, C, D)
 INSCRIPCIONES (Sid, Término, Sec_no)

Especifique y ejecute las siguientes consultas utilizando el intérprete de RA en el Esquema de la base de datos GRADEBOOK .

- a. Recuperar los nombres de los estudiantes inscritos en la clase de Autómatas durante el otoño de 2009 trimestre.
- segundo. Recupere los valores Sid de los estudiantes que se han inscrito en CSc226 y CSc227.
- C. Recupere los valores Sid de los estudiantes que se han inscrito en CSc226 o CSc227.

- re. Recupera los nombres de los estudiantes que no se han inscrito en ninguna clase.
- mi. Recuperar los nombres de los estudiantes que se han matriculado en todos los cursos del Mesa CATALOG .

8.37. Considere una base de datos que consta de las siguientes relaciones.

PROVEEDOR (Sno, Sname)
 PART (Pno, Pname)
 PROYECTO (Jno, Jname)
 SUMINISTRO (Sno, Pno, Jno)

La base de datos registra información sobre proveedores, piezas y proyectos y incluye una relación ternaria entre proveedores, piezas y proyectos. Esta La relación es una relación de muchos-muchos-muchos. Especifique y ejecute lo siguiente realizar consultas utilizando el intérprete de RA.

- a. Recupere los números de pieza que se suministran para exactamente dos proyectos.
- segundo. Recuperar los nombres de los proveedores que suministran más de dos piezas a proyecto 'J1'.
- C. Recupere los números de pieza proporcionados por cada proveedor.
- re. Recupere los nombres de los proyectos proporcionados únicamente por el proveedor 'S1'.
- mi. Recuperar los nombres de los proveedores que suministran al menos dos piezas diferentes. cada uno a al menos dos proyectos diferentes.

8.38. Especifique y ejecute las siguientes consultas para la base de datos en el ejercicio 5.16 utilizando el intérprete de RA.

- a. Recupere los nombres de los estudiantes que se han inscrito en un curso que utiliza un libro de texto publicado por Addison-Wesley-Longman.
- segundo. Recuperar los nombres de los cursos en los que se ha cambiado el libro de texto en menos una vez.
- C. Recuperar los nombres de los departamentos que adoptan los libros de texto publicados por Addison-Wesley únicamente.
- re. Recuperar los nombres de los departamentos que adoptan libros de texto escritos por Navathe y publicado por Addison-Wesley.
- mi. Recuperar los nombres de los estudiantes que nunca han usado un libro (en un curso) escrito por Navathe y publicado por Addison-Wesley.

8.39. Repita los ejercicios de laboratorio 8.34 a 8.38 en cálculo relacional de dominios (DRC) utilizando el intérprete de DRC.

Bibliografía seleccionada

Codd (1970) definió el álgebra relacional básica. Date (1983a) analiza las combinaciones externas. El trabajo sobre la extensión de las operaciones relacionales es discutido por Carlis (1986) y Ozsoyoglu y col. (1985). Cammarata y col. (1989) amplía la integridad del modelo relacional restricciones y uniones.

Codd (1971) introdujo el lenguaje Alpha, que se basa en conceptos de tupla cálculo relacional. Alpha también incluye la noción de funciones agregadas, que va más allá del cálculo relacional. La definición formal original de cálculo relacional fue dado por Codd (1972), que también proporcionó un algoritmo que transforma cualquier tupla expresión de cálculo relacional al álgebra relacional. El QUEL (Stonebraker et al., 1976) se basa en el cálculo relacional de tuplas, con cuantificadores existenciales implícitos, pero no cuantificadores universales, y se implementó en el sistema INGRES como un idioma comercialmente disponible. Codd definió la completitud relacional de un lenguaje de consulta guage significa al menos tan poderoso como el cálculo relacional. Ullman (1988) describe un prueba formal de la equivalencia del álgebra relacional con las expresiones seguras de cálculo relacional de tuplas y dominios. Abiteboul y col. (1995) y Atzeni y deAntonellis (1993) dan un tratamiento detallado de los lenguajes relacionales formales.

Aunque las ideas de cálculo relacional de dominio se propusieron inicialmente en el QBE lenguaje (Zloof, 1975), el concepto fue definido formalmente por Lacroix y Pirotte (1977a). La versión experimental del sistema Query-By-Example se describe en Zloof (1975). El ILL (Lacroix y Pirotte, 1977b) se basa en cálculos relacionales de dominio. Whang y col. (1990) amplía QBE con cuantificadores universales. Consulta visual idiomas, de los que QBE es un ejemplo, se proponen como medio de consulta bases de datos; conferencias como la Conferencia de trabajo de sistemas de bases de datos visuales (por ejemplo, Arisawa y Catarci (2000) o Zhou y Pu (2002)) presentan una serie de propuestas als para dichos lenguajes.