

5. *SELECT*. Cada grupo en el resultado del paso 4 genera una sola fila de resultado, como sigue. En primer lugar, se extraen del grupo el número de parte, el peso, el color y la cantidad máxima. En segundo lugar, el peso se convierte a gramos. En tercer lugar, se insertan las cadenas "Peso en gramos =" y "Cant. máx. enviada =" en los lugares apropiados de la fila.
6. *ORDER BY*. El resultado del paso 5 se ordena de acuerdo con las especificaciones de la cláusula *ORDER BY* (ordenar por), con lo cual se obtiene el resultado final.

Debe entenderse que el algoritmo recién descrito se presenta como una explicación puramente conceptual de la forma como se evalúa la proposición *SELECT* (seleccionar). Sin duda es correcta, en cuanto a que producirá con certeza el resultado correcto; de hecho, puede tomarse como definición de cuál debe ser el resultado (en otras palabras, sirve como definición de la semántica de la proposición *SELECT* original). No obstante, si se le ejecutara en realidad, lo más probable es que fuera bastante ineficiente. Por ejemplo, no sería muy conveniente, por razones de espacio de almacenamiento y también de tiempo de ejecución, la construcción del producto cartesiano sugerida en el paso 1. Consideraciones como ésta son precisamente las que justifican la necesidad de un optimizador en los sistemas relacionales. De hecho, el objetivo del optimizador puede definirse como encontrar un procedimiento que produzca el mismo resultado producido por el algoritmo conceptual antes descrito pero que sea más eficiente en términos de espacio o de tiempo (o de ambos, si es posible).

Por último, acerca del ejemplo de consulta en sí: la proposición *SELECT* mostrada es desde luego bastante compleja, pero pensemos en todo el trabajo que realiza. Un programa convencional para efectuar la misma tarea, escrito en un lenguaje como COBOL, podría ocupar con facilidad nueve páginas, en vez de sólo nueve renglones, y la labor requerida para poner en funcionamiento ese programa sería bastante mayor que la implicada en la construcción de la versión SQL mostrada. En la práctica, por supuesto, la mayor parte de las consultas serán de todos modos más sencillas que ésta.

EJERCICIOS

Todos estos ejercicios utilizan la base de datos de proveedores, partes y proyectos (véanse los ejercicios del capítulo 5). En casi todos (excepto los números 6.37 a 6.40 y 6.42, como puede verse) se pide al lector que escriba una proposición o conjunto de proposiciones en SQL para la operación indicada. Por conveniencia repetimos aquí la estructura de la base de datos:

```
S ( S#, SNOMBRE, SITUACIÓN, CIUDAD )
P ( P#, PNOMBRE, COLOR, PESO, CIUDAD )
J ( J#, JNOMBRE, CIUDAD )
SPJ ( S#, P#, J#, CANT )
```

Dentro de cada subsección, los ejercicios se presentan en orden aproximado de dificultad creciente. Se recomienda intentar al menos algunos de los más fáciles en cada grupo. Los números 34 a 40 son bastante difíciles.

Consultas sencillas

- 6.1 Obtener los detalles completos de todos los proyectos.
- 6.2 Obtener los detalles completos de todos los proyectos de Londres.
- 6.3 Obtener los números de los proveedores que suministran partes al proyecto J1, ordenados por número de proveedor.
- 6.4 Obtener todos los envíos en los cuales la cantidad está en el intervalo de 300 a 750 inclusive.
- 6.5 Obtener una lista de todas las combinaciones parte-color/parte-ciudad, eliminando todas las parejas color/ciudad repetidas.

Reuniones

- 6.6 Obtener todas las tripletas número de proveedor/número de parte/ número de proyecto tales que el proveedor, la parte y el proyecto indicados estén todos en la misma ciudad (cosituados).
- 6.7 Obtener todas las tripletas número de proveedor/número de parte/ número de proyecto tales que el proveedor, la parte y el proyecto indicados no estén todos cosituados.
- 6.8 Obtener todas las tripletas número de proveedor/número de parte/ número de proyecto tales que el proveedor, la parte y el proyecto indicados estén todos en diferente ciudad.
- 6.9 Obtener los números de las partes suministradas por algún proveedor de Londres.
- 6.10 Obtener los números de las partes suministradas por un proveedor de Londres a un proyecto en Londres.
- 6.11 Obtener todas las parejas de nombres de ciudad tales que un proveedor de la primera ciudad suministre partes a un proyecto en la segunda ciudad.
- 6.12 Obtener los números de las partes suministradas a un proyecto por un proveedor situado en la misma ciudad que el proyecto.
- 6.13 Obtener los números de los proyectos a los cuales suministra partes por lo menos un proveedor situado en una ciudad distinta.
- 6.14 Obtener todas las parejas de números de parte tales que algún proveedor suministre las dos partes indicadas.

Funciones de agregados

- 6.15 Obtener el número total de proyectos a los cuales suministra partes el proveedor S1.
- 6.16 Obtener la cantidad total de la parte P1 suministrada por el proveedor S1.
- 6.17 Para cada parte suministrada a un proyecto, obtener el número de parte, el número de proyecto y la cantidad total correspondiente.
- 6.18 Obtener los números de las partes suministradas a algún proyecto tales que la cantidad promedio suministrada sea mayor que 320.

Diversas

- 6.19 Obtener todos los envíos para los cuales la cantidad no sea nula.
- 6.20 Obtener números de proyecto y ciudades en los cuales la segunda letra del nombre de la ciudad sea una "o".

Subconsultas

- 6.21 Obtener los nombres de los proyectos a los cuales suministra partes el proveedor S1.
- 6.22 Obtener los colores de las partes suministradas por el proveedor S1.
- 6.23 Obtener los números de las partes suministradas a cualquier proyecto en Londres.
- 6.24 Obtener los números de los proyectos donde se utilice al menos una de las partes suministradas por el proveedor S1.
- 6.25 Obtener los números de los proveedores que suministren por lo menos una de las partes suministradas por al menos uno de los proveedores que suministran por lo menos una parte roja.
- 6.26 Obtener los números de los proveedores cuya situación sea inferior a la del proveedor S1.
- 6.27 Obtener los números de los proyectos cuya ciudad sea la primera en la lista alfabética de las ciudades donde hay proyectos.
- 6.28 Obtener los números de los proyectos a los cuales se suministre la parte P1 en una cantidad promedio mayor que la cantidad máxima en la cual se suministra alguna parte al proyecto J1.
- 6.29 Obtener los números de los proveedores que suministren la parte P1 a algún proyecto en una cantidad mayor que la cantidad promedio enviada de la parte P1 para ese proyecto.

EXISTS (EXISTE)

- 6.30 Repetir el ejercicio 6.23 utilizando EXISTS en la solución.
- 6.31 Repetir el ejercicio 6.24 utilizando EXISTS en la solución.
- 6.32 Obtener los números de los proyectos a los cuales no suministra ninguna parte roja ninguno de los proveedores de Londres.
- 6.33 Obtener los números de los proyectos para los cuales S1 es el único proveedor.
- 6.34 Obtener los números de las partes suministradas a todos los proyectos en Londres.
- 6.35 Obtener los números de los proveedores que suministren la misma parte a todos los proyectos.
- 6.36 Obtener los números de los proyectos a los cuales se suministren por lo menos todas las partes suministradas por el proveedor S1.

En los siguientes cuatro ejercicios (6.37 a 6.40), convertir la proposición SELECT de SQL mostrada a su equivalente en español.

```
6.37  SELECT DISTINCT J#
      FROM   SPJ SPJX
      WHERE  NOT EXISTS
            ( SELECT *
              FROM   SPJ SPJY
              WHERE  SPJY.J# = SPJX.J#
              AND    NOT EXISTS
                    ( SELECT *
                      FROM   SPJ SPJZ
                      WHERE  SPJZ.P# = SPJY.P#
                      AND    SPJZ.S# = 'S1' ) ) ;
```

```
6.38  SELECT DISTINCT J#
      FROM   SPJ SPJX
      WHERE  NOT EXISTS
            ( SELECT *
              FROM   SPJ SPJY
              WHERE  EXISTS
                    ( SELECT *
                      FROM   SPJ SPJA
```

```

WHERE SPJA.S# = 'S1'
AND SPJA.P# = SPJY.P# )
AND NOT EXISTS
( SELECT *
FROM SPJ SPJB
WHERE SPJB.S# = 'S1'
AND SPJB.P# = SPJY.P#
AND SPJB.J# = SPJX.J# ) ) ;

```

```

6.39 SELECT DISTINCT J#
FROM SPJ SPJX
WHERE NOT EXISTS
( SELECT *
FROM SPJ SPJY
WHERE EXISTS
( SELECT *
FROM SPJ SPJA
WHERE SPJA.P# = SPJY.P#
AND SPJA.J# = SPJX.J# )
AND NOT EXISTS
( SELECT *
FROM SPJ SPJB
WHERE SPJB.S# = 'S1'
AND SPJB.P# = SPJY.P#
AND SPJB.J# = SPJX.J# ) ) ) ;

```

```

6.40 SELECT DISTINCT J#
FROM SPJ SPJX
WHERE NOT EXISTS
( SELECT *
FROM SPJ SPJY
WHERE EXISTS
( SELECT *
FROM SPJ SPJA
WHERE SPJA.S# = SPJY.S#
AND SPJA.P# IN
( SELECT P#
FROM P
WHERE COLOR = 'Rojo' )
AND NOT EXISTS
( SELECT *
FROM SPJ SPJB
WHERE SPJB.S# = SPJY.S#
AND SPJB.J# = SPJX.J# ) ) ) ) ;

```

Unión

6.41 Construir una lista ordenada de todas las ciudades en las cuales esté situado por lo menos un proveedor, una parte o un proyecto.

6.42 Mostrar el resultado de la siguiente selección:

```

SELECT P.COLOR
FROM P
UNION
SELECT P.COLOR
FROM P ;

```

Operaciones de actualización

6.43 Cambiar a gris el color de todas las partes rojas.

6.44 Eliminar todos los proyectos para los cuales no haya envíos.