13テーブルの結合(31)

テーブルの結合(31)

これまでの検索は、少数の例外を除いて、基本的には一つのテーブルを検索するもの だった。しかし、二つ以上のテーブルを組み合わせなければ、検索が出来ないことは 決して珍しくない。例えば、ある従業員の所属部門を知りたいとしよう。ところが、テーブ ル「従業員」には、所属部門のコードは載っているが、部門の名前自体は載っていな い。部門の名前を調べるためには、テーブル「部門」を用いて、部門番号 から部門名を調べなければならない。要するに、ある従業員の所属部門を調べるために は、「従業員」と「部門」という二つのテーブルが必要であり、二つのテーブ ルを見比べて、双方の「部門番号」のカラムが等しい行を選び出して、その中から 適当な項目を選んで新しいテーブルを作成するという操作が必要となる。こうした操作 を、SQLでは、「テーブルのジョイン」と呼ぶ。「ジョイン」は、テーブルからテーブル を作り出す、テーブルについて閉じた演算だが、その元になっているのは、複数のテー ブルの積をつくり（これは、from句に複数のテーブルを置くことによって可能となる）、その大きなテーブルから、条件にあう行のみをselectする（もちろんselectの条件は where句に置かれる）という操作である。SQLでジョインを実現する為には、こうした 操作を忠実に定式化すればいいのである。

まず、SQLでのジョインの実例を見てみよう。次の例は、横浜市から通勤している従業員の名前と所属部門を検索するものである。

例 31: 横浜市在住の従業員の所属部門名を検索（ジョイン）

```
select substring(部門.部門名,1,10),substring(従業員.氏名,1,6),substring(従業員.住所,1,6)
from 部門,従業員
where 従業員.住所 like '%横浜市%'
and 従業員.部門番号 = 部門.部門番号
```

部門名	氏名	住所
土浦工場生産部門ライン2	坂上次郎	横浜市
つくば工場生産部門ライン1	三木和正	横浜市
下妻工場生産部門ライン１	桜谷由香里	横浜市

substringというのは、部分文字列を返す関数で、selectの項目名のリストの部分に 置くことが出来る。この関数の三つの引き数は、一番目がもとの文字列で、二番目、 三番目の引き数が、それぞれ部分文字列が開始、終了される位置を表している。 次の例を見れば、この関数の働きは明かであろう。

```
substring('abcdefghij',1,6) ===> 'abcdef'
substring('abcdefghij',1,5) ===> 'abcde'
substring('abcdefghij',1,4) ===> 'abcd'
```

```
substring('abcdefghij',2,6) ===> 'bcdef'
substring('abcdefghij',3,6) ===> 'cdef'
substring('abcdefghij',4,6) ===> 'def'
```

長い部門名や名前がある場合はこの関数によって名前の一部が切り捨てられる。この例 でのsubstringの使用は、単に出力を一行に収めようとする為のもので、ジョインの操 作とは何の関わりもない。これ以降の例では、プログラムの見やすさの為に、select文 の中でのsubstringなどの関数を省略して示す。