

18selectのネスティング(36)(37)(38)

selectのネスティング(36)(37)(38)

テーブルのジョインを行うには、もう一つの方法がある。それは、以前の例21で簡単に見てきたが、select文の内部で再びselect文を用いる方法である。

/* 例 21 */

```
select 氏名,住所,年齢
from 従業員
where 住所 in (select 市町村名 from 茨城神奈川県 where 県名='神奈川県')
```

ここでのselectのネスティングでは、次の

```
where 項目名 in ( select 項目名 from ..... )
```

という形が特徴的である。ここでの二つの項目は、もちろん、同じタイプに属さなければならない。加えて、内側のselect文は、一つの項目のみを返すものでなければならないことに注意されたい。逆に、こうした要件が満たされていれば、in 以外の比較演算子を置くことも可能である。例えば、

```
where 項目名 = ( select 項目名 from ..... )
where 項目名 < ( select 項目名 from ..... )
where 項目名 > ( select 項目名 from ..... )
.....
```

といった形式でのサブ・クエリーが可能である。

次の例は、例32とほとんど同じ、横浜市在住の従業員の所属部門を検索するものである。

例 36: 横浜市在住の従業員の所属部門名を検索（サブ・クエリー）

```
select distinct 部門名 from 部門
where 部門番号 in
(select 部門番号 from 従業員 where 住所 like '%横浜%')
```

部門名
つくば工場生産部門ライン1
下妻工場生産部門ライン 1
土浦工場生産部門ライン2

概念的には、最も内側のselect文から実行が行われる。この例では、従業員という テーブルから、横浜市という文字列を含む住所が選ばれ、その部門番号が、返さ れる。通常は、select文の出力は、テーブルを形成するのだが、このselectのネスティ ングでは、内側のselectの出力は、まずリストとして解釈される。ここでは、内側の select文の出力は、条件を満たす部門番号のリストを形成して、where句の中で、 今度は、「部門」のテーブル中の部門番号の項目と比較される。ここでのdistinctは、あい異なる部門のみを出力する働きを持つ。横浜から通勤する従業員が2人いる部門の名前を2度カウントする必要はないからである。

次の例は、例31に対応している。

例 37: 横浜市在住の従業員の所属部門名を検索完全修飾（サブ・クエリー）

```
select distinct 部門.部門名 from 部門,従業員
where 部門.部門番号 in
(select 部門番号 from 従業員 where 住所 like '%横浜%')
```

部門名
土浦工場生産部門ライン2
つくば工場生産部門ライン1
下妻工場生産部門ライン 1

次の例は、例33での三つのテーブルのジョインに対応している。selectのネストも 三重になっていることに注目せよ。

例 38: 単価100円以下の仕入を実現した業者の親会社名を検索（サブ・クエリー）

```
select 親会社名
from 親会社
where 親会社番号 in
( select 親会社番号
from 業者
where 業者番号 in
( select 業者番号
from 仕入
where 単価 <=100
)
)
```

親会社名
三友物産
三友化学
住菱商事